MASTER EN SCIENCES INFORMATIQUES

Etude des rôles des fournisseurs SaaS et Cloud dans la sécurité& la vie privée, la disponibilité et la scalability

Robinet, Frédéric

Award date:
2013

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'Informatique
Année académique 2012–2013

Etude des rôles des fournisseurs SaaS et Cloud dans la sécurité & la vie privée, la disponibilité et la scalability

Frédéric ROBINET

Promoteur : _____ (Signature pour approbation du dépôt - REE art. 40)
PhD Philippe THIRAN

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques.

Résumé

Dans ce travail, nous étudions les deux rôles d'une solution de type SaaS : le Cloud Provider et le SaaS Provider. Nous commençons par définir le Cloud Computing, ses fonctionnalités, ses modèles de déploiement et de service. Nous présentons ensuite les deux acteurs identifiés et la relation qui les unit, l'Utility Computing.

Nous abordons ensuite trois thèmes : la sécurité & la vie privée, la scalability et la disponibilité. Pour chacun de ces thèmes, nous proposons une structure de décomposition en sous-thèmes et identifions une série d'outils qui permettent d'atteindre l'objectif, en indiquant à quel acteur l'implémentation revient.

En synthèse, nous proposons un questionnaire d'évaluation d'une solution de type SaaS, se basant sur les trois thèmes. Ce questionnaire permet d'obtenir un score exprimé en pourcentage de couverture pour chaque élément de décomposition des thèmes traités.

Remerciements

Je souhaite adresser mes remerciements à l'ensemble des personnes qui ont contribué à la réalisation de ce travail. Je remercie en particulier M. Philippe Thiran, promoteur de ce mémoire, pour ses conseils et sa disponibilité qui ont permis d'aboutir à ce résultat. J'adresse également mes remerciements à l'ensemble des professeurs et assistants de l'université, qui m'ont enseigné les compétences indispensables à cette entreprise, ainsi qu'à Benjamine Lurquin pour l'encadrement administratif qu'elle m'a apporté.

Je tiens également à exprimer ma reconnaissance envers ma famille, qui m'a encouragé et soutenu tout au long de ces études.

Table des matières

Chapitre 1

Introduction

SaaS et Cloud sont deux termes omniprésents dans l'actualité informatique d'aujourd'hui. Le déploiement marketing mis en place sur ces sujets est tel que la définition et la compréhension par la communauté IT sont loin de faire l'unanimité. Certains y voient une révolution, d'autres une évolution et d'autres encore, comme Richard Stallman, un piège du marketing sur les libertés individuelles [18].

Cette ambiguïté nous impose de définir ces deux termes. Dans la suite de ce document, nous utiliserons le terme "Cloud" pour représenter le matériel et les logiciels systèmes qui sont situés dans des datacenters. Le terme "SaaS" représentera quant à lui la couche des applications construites sur les infrastructures Cloud.

Les solutions de type Software as a Service (SaaS) ont envahi notre quotidien : boîtes mail, CRM, ERP, outils de visio-conférences, outils collaboratifs, etc. De nouvelles applications sont créées chaque jour pour enrichir les gammes existantes, offrir de nouveaux produits qui prennent la forme de services, payants ou non.
L'utilisation du terme SaaS se retrouve dans une grande quantité d'offres. Mais que couvre précisément le SaaS ? Le SaaS se limite-t-il à une externalisation des applications et des données de l'entreprise, moyennant un abonnement ? Le SaaS et le Cloud sont-ils absolument liés entre eux ?

Pour le Cloud, d'énormes investissements ont été réalisés par l'ensemble des grands acteurs informatiques comme IBM, Oracle, Microsoft, Google et Amazon, dans la construction de datacenters répartis tout autour du globe. Chaque acteur propose aujourd'hui sa solution spécifique, avec sa propre technologie et différents niveaux de services.
Mais quelles sont les différences entre ces niveaux de services ? Qu'est-ce que le Cloud peut apporter à une entreprise ?

A partir des définitions du Cloud et du SaaS, nous avons identifié deux rôles : le SaaS Provider, qui fournit aux utilisateurs finaux l'application, et le Cloud Provider, qui fournit au SaaS Provider les ressources informatiques nécessaires au bon fonctionnement de l'application. Cette relation entre le Cloud Provider et le SaaS Provider sera nommée "Utility Computing", comme indiqué sur la Figure 1.1. Ces deux rôles peuvent bien entendu être tenus par une seule et même entreprise, comme c'est par exemple le cas avec Google qui exploite son infrastructure Cloud pour proposer différents services, comme Gmail.

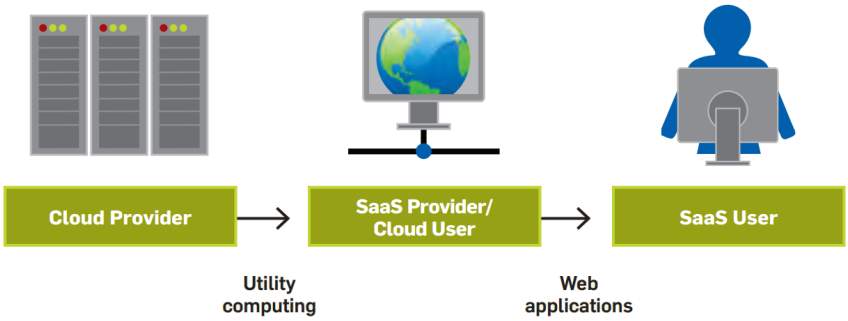


FIGURE 1.1 – L'Utility Computing [7]

Ce travail est issu de l'article de référence [7], à partir duquel nous avons commencé par identifier les grandes thématiques comme la performance, l'usability, la sécurité, l'auditabilité, etc.
Nous avons commencé par définir le Cloud Computing, avec ses fonctionnalités, ses modèles de déploiement et de services. Nous avons complété ce chapitre 2 par la présentation des deux rôles, le SaaS Provider et le Cloud Provider, et la relation qui les unit, l'Utility Computing. Nous nous sommes ensuite focalisés sur trois sujets : la sécurité et la vie privée (chapitre 3), la scalability (chapitre 4) et la disponibilité (chapitre 5).
Pour chacun de ces sujets, nous avons compulsé un ensemble de livres, d'articles et de publications scientifiques pour dégager une structure de décomposition à plusieurs niveaux. Pour chaque sous-thème identifié, nous avons sélectionné une série d'outils ou technologies permettant de répondre à cette attente, en indiquant pour chaque solution l'acteur qui a la charge de l'implémenter.

Sécurité & vie privée Nous avons sélectionné la sécurité et la vie privée car elles représentent à l'heure actuelle un des obstacles majeurs à l'adoption du Cloud Computing et du SaaS, comme le révèle l'enquête de KPMG de 2010 sur le Cloud Computing [22] dans la Figure 1.2. Cette préoccupation a pour origine le transfert de la responsabilité des données et services qui étaient auparavant gérés par l'entreprise et qui se retrouve maintenant répartie entre le Cloud Provider et le SaaS Provider.

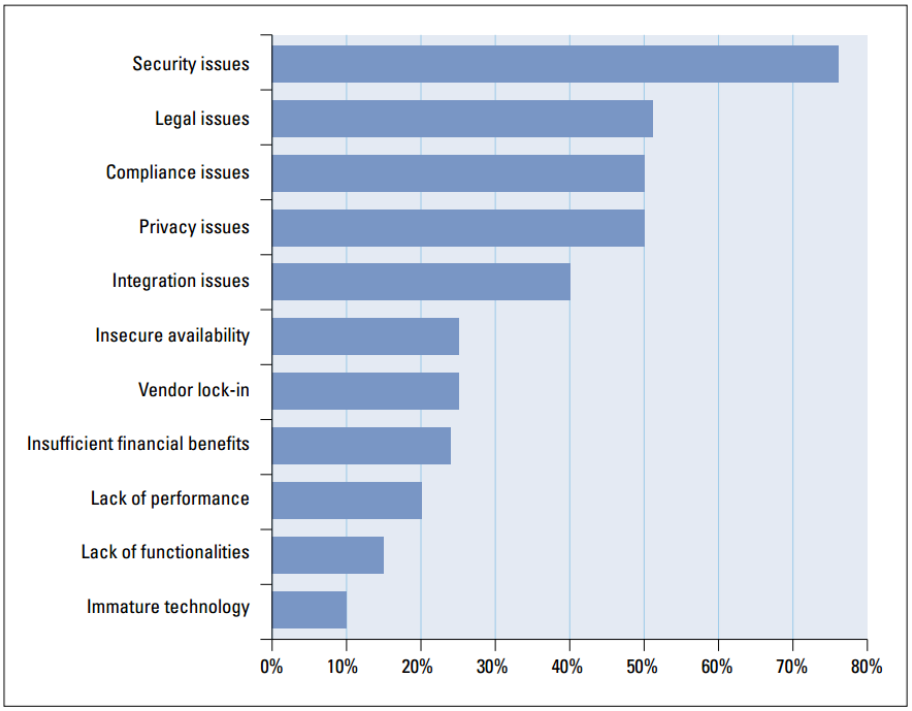


FIGURE 1.2 – Les principales préoccupations des responsables IT concernant le Cloud Computing [22]

Notre analyse de la sécurité nous a orienté vers une scission de ce thème en deux parties : la partie infrastructure - qui regroupe la sécurité au niveau réseau, hôtes et applications - et la partie données.

Scalability La scalability est une des pierres angulaires du Cloud Computing, ainsi qu'un de ses nombreux défis. Elle est régulièrement citée comme une de ses caractéristiques principales, sous le terme "élasticité". Mais elle implique bien davantage que l'ajout d'instances virtuelles et l'attribution de ressources informatiques : elle nécessite une réflexion et une architecture orientées vers cet objectif, de façon à optimiser au mieux les ressources et leur allocation. Nous analyserons dans ce chapitre les différents aspects de la scalability et les outils mis en place par le SaaS Provider et le Cloud Provider pour l'atteindre.

Disponibilité La troisième thématique abordée, la disponibilité, nous a été inspirée par les différentes pannes de service ayant affecté différents fournisseurs de Cloud Computing, comme ce fut le cas de Microsoft avec sa panne mondiale du 29 février 2012 [29] ou plus récemment, la panne du service ELB d'Amazon, survenue le 24 décembre 2012, qui a affecté les performances de grands sites comme Netflix. Ces interruptions de service ont différentes origines : cause naturelle (un orage sur Dublin en 2011), bugs logiciels, problème matériel, coupure d'électricité, etc. Nous verrons dans ce chapitre les stratégies mises en oeuvre à

différents niveaux pour permettre de conserver une disponibilité maximale.

Au terme de chacun de ces chapitres, nous établirons un tableau récapitulatif reprenant les différentes catégories de chaque thème, avec les outils utilisés et l'acteur devant les implémenter.

En synthèse, nous tenterons de développer un système d'évaluation du niveau de maturité d'une solution, en attribuant une note au niveau de fonctionnalité du SaaS Provider et du Cloud Provider. Nous appliquerons ensuite cette méthode à un exemple.

Ce travail est une première tentative de définition de la répartition des rôles des différents acteurs d'une solution SaaS. L'objectif est de permettre aux décideurs informatiques de comprendre les implications du passage au Cloud Computing, de permettre d'évaluer les différentes solutions proposées et de déterminer l'acteur responsable de l'implémentation d'une fonctionnalité.

Chapitre 2

Cloud Computing

Le Cloud Computing (en français, informatique dans le nuage ou informatique virtuelle) est un terme assez difficile à définir étant donné que cette appellation est extrêmement exploitée comme argument commercial. Microsoft, Google, Amazon, Apple, Sun, IBM, ... sont des fournisseurs de solutions Cloud mais chacun met en avant sa propre technologie et ses fonctionnalités spécifiques. Le Cloud Computing ne doit donc pas être perçu comme une technologie bien précise mais plutôt comme un concept, généralement accompagné d'un nouveau Business Model, s'appuyant sur ces nouvelles technologies.

Dans la suite de ce document, nous définirons le Cloud Computing comme *"un modèle permettant un accès pratique, permanent et à la demande par le réseau à un ensemble de ressources informatiques configurables, dont les capacités peuvent être étendues et activées rapidement avec un effort minimal"* [33]. Il s'agit donc d'une mutualisation des ressources informatiques, auxquelles les utilisateurs ont accès via le réseau, généralement Internet. Cette mutualisation va permettre de centraliser les traitements informatiques qui étaient précédemment localisés sur des serveurs locaux ou sur le poste de l'utilisateur [33].

Le Cloud Computing a donc deux composantes [6] :
- le matériel et les logiciels systèmes, qui sont situés dans des datacenters. Cet ensemble sera nommé *Cloud* dans la suite de ce document ;
- les applications, qui sont délivrées comme des services via le réseau.

Le concept du Cloud Computing en lui-même n'est pas nouveau : Internet s'est construit sur base de serveurs partagés permettant l'échange de mails, l'accès à des pages Web et d'autres services. Toutefois, le Cloud Computing se présente ici comme un pas supplémentaire vers la virtualisation continue des systèmes informatiques : l'évolution des capacités de calcul et des bandes passantes permet aujourd'hui d'envisager le partage des ressources informatiques entre un grand nombre d'utilisateurs et de satisfaire leurs besoins respectifs en terme de ressources informatiques [49]. Dans [15], les auteurs identifient trois raisons favorisant l'émergence du Cloud Computing :

1. Les infrastructures informatiques existantes dans les entreprises ne sont plus adaptées à la quantité de données, de transactions et de dispositifs numériques qui ont explosé ces dernières années. Dans [17], les auteurs estiment la quantité des données numériques créées dans le monde à 1,2 zettaoctets en 2010, 2,8 zettaoctets en 2012 et prévoient 40 zettaoctets en 2020 ;
2. La bande passante des réseaux et les capacités de stockage ne savent pas faire face à la croissance exponentielle du nombre d'abonnés et des services de communications ;
3. Les pics de consommation et l'inefficacité de la fourniture des ressources provoquent une pression sur les systèmes énergétiques.

Ces facteurs ont favorisé l'éclosion du Cloud Computing, qui a débuté en 2006 avec la solution EC2 d'Amazon. Depuis d'autres grandes sociétés (IBM, Microsoft, Google, etc) se sont également lancées dans l'aventure, avec des offres et des technologies spécifiques.

2.1 Description

Le National Institute of Standards and Technology [33], dans sa définition du Cloud Computing, décrit cinq fonctionnalités, quatre modèles de déploiement et trois modèles de service.

2.1.1 Cinq fonctionnalités

Pour répondre à la définition du NIST, un Cloud va devoir implémenter les fonctionnalités suivantes :

- **Elasticité** : le consommateur peut étendre les capacités de son environnement Cloud de manière élastique, éventuellement de façon automatique, pour répondre à ses attentes. Cette caractéristique va permettre au consommateur d'adapter les ressources informatiques à la demande : lorsque celle-ci augmente ou diminue, il adapte la quantité de ressources allouées au service. Le consommateur ne doit donc plus investir dans une infrastructure conséquente pour répondre aux pics de charge ou risquer le manque de capacité : le risque d'*under-provisioning* ou d'*over-provisioning* est donc évité et il peut consacrer ses ressources financières et humaines à son coeur de métier, le service qu'il propose. Cette optimisation dynamique est illustrée à la Figure 2.1.

- **Accès par le réseau** : le consommateur accède à ses services Cloud au travers d'un réseau, généralement Internet. De nombreuses plates-formes (smartphones, ordinateurs portables et fixes, etc) peuvent donc y accéder aisément.

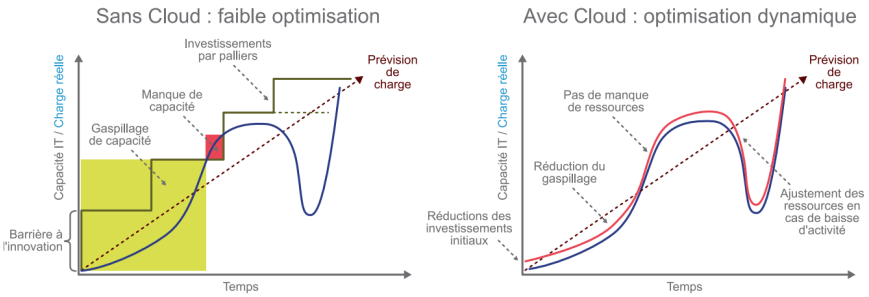


FIGURE 2.1 – Cloud Computing - L'optimisation par l'élasticité [38]

– **Mise en commun des ressources** : un Cloud est partagé entre plusieurs consommateurs avec une allocation dynamique des ressources en fonction des demandes de chacun. Avec ce concept, le consommateur ne sait pas précisément où se trouvent les ressources mises à sa disposition. Pour parvenir à cette fonction, le Cloud fait appel à la virtualisation, qui va permettre de partager une ressource physique comme un processeur ou de la mémoire entre différents consommateurs. Ce partage va concerner les capacités de calcul, de stockage, de réseau et de mémoire.

– **'Self-service' à la demande** : les ressources mises à la disposition du consommateur peuvent être réduites ou étendues par le consommateur lui-même, sans intervention du fournisseur. Le fournisseur met à la disposition du consommateur un ensemble d'outils, sous forme d'applications ou d'API, pour lui permettre d'augmenter les ressources qu'il consomme. Un exemple de ce type de fonctionnalités est la Management Console d'Amazon Web Services, qui donne accès à tous les services de la plate-forme : création de nouvelles instances virtuelles, de nouvelles bases de données, de réseaux virtuels, etc.

– **Monitoring** : un Cloud offre des outils de monitoring à différents niveaux, qui sont disponibles pour le consommateur et le fournisseur. Ces outils vont permettre d'une part d'optimiser l'utilisation des ressources globales et d'autre part de mesurer précisément les ressources qu'un consommateur utilise, en vue d'une facturation.

2.1.2 Quatre modèles de déploiement

Une infrastructure de Cloud Computing peut se présenter sous la forme de différents modèles de déploiement [58] :

– **Cloud public (public Cloud)** : un Cloud public est la propriété d'une entreprise (Google, Microsoft, etc), d'une université ou d'un gouvernement et est géré par eux. Son accès est public mais nécessite une inscription. Ce type de Cloud offre un ensemble de services standardisés avec une tarification à l'usage (price-per-use) ou gratuitement : business process, appli-

cations, services d'infrastructure, etc. Ce type d'infrastructure offre une certaine standardisation, une grande flexibilité et un temps de déploiement court, tout en préservant le capital du consommateur [15].

- **Cloud privé (private Cloud)** : dans cette configuration, le Cloud est mis à la disposition exclusive d'une seule entreprise, dont les différents départements vont être les consommateurs. L'infrastructure est alors soit la propriété de l'organisation, soit celle d'un partenaire, soit un mix des deux [33]. Cette infrastructure offre certains des avantages du Cloud public, comme la flexibilité et la standardisation des bonnes pratiques, tout en permettant à l'entreprise de conserver le contrôle sur le matériel, sur la localisation et la sécurisation des données. Toutefois, un capital très important est nécessaire au départ pour la constitution du (ou des) datacenter(s).

- **Cloud communautaire (community Cloud)** : dans ce modèle de déploiement, l'infrastructure du Cloud est mise à la disposition exclusive d'une communauté de consommateurs provenant d'entreprises différentes qui partagent un même centre d'intérêt. L'infrastructure appartient alors soit à un des membres, soit à un tiers ou une combinaison des deux.

- **Cloud hybride (hybrid Cloud)** : un Cloud hybride est une composition d'un Cloud privé et d'un Cloud public, liés par une technologie standard ou propriétaire, de façon à permettre la portabilité des applications de l'un vers l'autre [33]. Ce modèle est avant tout un Cloud privé qui permet à son propriétaire d'utiliser un Cloud public quand et où il est logique de le faire [24]. L'objectif premier de cette structure est de répondre à des besoins spécifiques de sécurité : l'entreprise peut décider elle-même quelles données sont gérées par son Cloud privé ou par le Cloud public, en fonction du niveau de confidentialité requis.

Certains éditeurs proposent des solutions permettant de modifier facilement le modèle de déploiement. Par exemple, Eucalyptus offre une compatibilité avec les API d'AWS, ce qui permet la portabilité du développement d'un environnement purement privé vers un hybride ou vers le modèle public.

2.1.3 Trois modèles de service

L'architecture à mettre en place pour fournir un service est composée de 9 couches :

1. Le réseau, qui va relier les différents serveurs et stockages entre eux, ainsi qu'exposer le service à l'extérieur ;
2. Le stockage, qui se compose de matériel (disques durs), d'un système de fichiers (GFS, etc) et de mécanismes de backup ;
3. Les serveurs physiques ;
4. La virtualisation, qui va être réalisée par un hyperviseur natif, aura pour rôle de distribuer les ressources physiques entre les différents systèmes

d'exploitation. Ex : Xen, ESX Server, etc ;

5. Le logiciel serveur, c'est-à-dire le système d'exploitation qui sera installé dans l'environnement virtualisé ;
6. Les bases de données ;
7. Le middleware, qui va permettre de créer une architecture SOA (Service Oriented Architecture) ;
8. Les runtimes, c'est-à-dire les librairies de composants nécessaires à l'application ;
9. L'application proprement dite, qui offre le service.

Dans une architecture classique, l'ensemble de ces couches est gérée et hébergée par l'entreprise. Dans les trois modèles de services du Cloud Computing, ces différentes couches sont prises successivement en charge par le Cloud Provider, comme illustré à la Figure 2.2 :

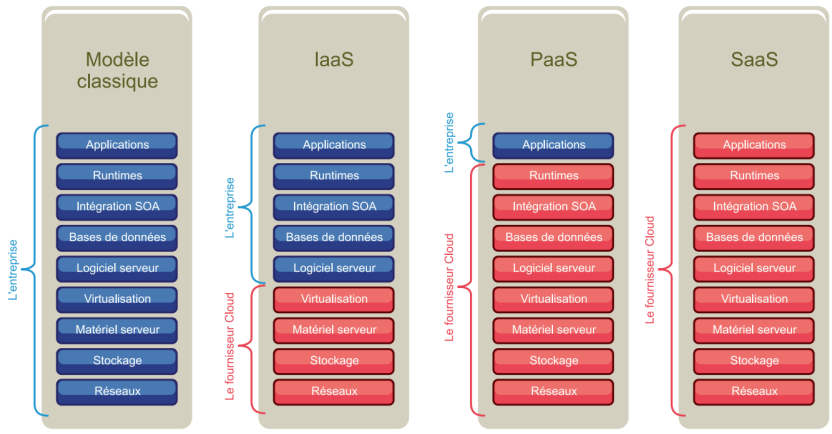


FIGURE 2.2 – Cloud Computing - Qui maintient quoi ? [38]

– L'**IaaS** (Infrastructure as a Service) est le modèle de service qui consiste pour le fournisseur à offrir une infrastructure informatique hébergée, sur laquelle le client pourra déployer des logiciels arbitraires, ce qui peut inclure le système d'exploitation, les runtimes et l'application [38]. Les machines se présentent sous la forme d'instances virtuelles, auxquelles l'accès est complet et sans restriction. Par exemple, une entreprise pourrait louer une instance pour y installer Linux et un serveur Apache et ensuite y déployer un site Web.
La différence par rapport à un hébergement classique est dans sa flexibilité : la configuration des instances (CPU, mémoire, espace disque, etc) peut être étendue ou réduite rapidement en fonction des besoins, avec une tarification variable en fonction de la consommation. Dans le cas d'un hébergement classique (comme OVH le propose), la configuration et les

coûts sont fixes, ce qui engendre généralement une sous-utilisation des ressources louées.
Les couches prises en charge sont donc le réseau, le stockage, les serveurs physiques et la virtualisation.

Une des premières entreprises à avoir fourni ce type de services est Amazon, qui a proposé au départ deux services [54] :
- le service EC2 : le développeur peut créer des instances virtuelles ayant des capacités particulières (OS, RAM, CPU, espace disque, etc) qu'il peut configurer finement et qu'il peut ensuite multiplier en fonction de ses besoins ;
- le service S3, qui permet de consommer l'espace de stockage dont l'utilisateur a besoin, et qu'il peut étendre à tout moment.

Depuis Amazon, d'autres sociétés ont également commencer à fournir du IaaS : VMWare, Parallels, etc.

- Le **PaaS** (Platform as a Service) est une couche supérieure au IaaS, dans le sens où elle consiste pour le fournisseur à offrir un environnement de développement et une infrastructure de déploiement : elle se compose donc d'un environnement d'exécution lié au langage de programmation (Ruby, Java, Python, .Net, etc.), de librairies de programmation, des outils de test et de monitoring, de bases de données, etc. Le consommateur développe ainsi ses applications en s'appuyant sur ces outils spécifiques du fournisseur.
  Il s'agit donc pour le fournisseur de mettre un middleware à disposition du consommateur. Ce middleware va assurer la gestion automatique de l'infrastructure et faciliter le déploiement des applications.
  Le principal désavantage du PaaS est que chaque fournisseur apporte sa propre technologie, avec un langage de développement de prédilection : Python et Java pour Google, .Net pour Windows Azure, Ruby pour Heroku, etc. Cette absence de standardisation des fonctions et API emprisonne le développeur dans la plateforme choisie et il devient donc dépendant du gestionnaire [54]. Des tentatives de standardisation sont actuellement en cours, comme l'IEEE Cloud Computing [8], mais aucune n'a encore abouti à l'heure actuelle.

  Comme exemple d'utilisation, nous pouvons citer le cas d'une entreprise qui chercherait à développer un site Web de ventes d'articles en se basant sur le framework vFabric de VMware, qui lui offre toute une série de fonctionnalités de développement spécifiques à cette architecture.
  Exemples de fournisseurs de PaaS : Windows Azure, Google App Engine, Force.com, etc.

- Le **SaaS** (Software as a Service) est la couche supérieure des modèles de service, qui consiste à mettre à disposition des logiciels via le Web. Les SaaS sont donc *"des applications construites sur les infrastructures Cloud (IaaS et PaaS)"* [54].
  Aujourd'hui, la majorité des services mis à disposition s'est concentrée sur les outils collaboratifs : mails, gestionnaire de relation client (CRM),

vidéo conférence, communications unifiées, travail collaboratif, outils de productivité (comme BaseCamp et HighRise), etc. Mais les services offerts ne cessent de s'étoffer et s'attaquent à d'autres segments, comme par exemple la finance, les ressources humaines, la logistique, etc.
Typiquement, une entreprise qui est à la recherche d'une plateforme de gestion de clientèle (CRM) pourrait s'orienter vers une solution comme SalesForce.

Il existe deux différences fondamentales entre les logiciels traditionnels et les SaaS [54] :

1. L'accès aux applications et aux données : dans une solution SaaS, l'application et les données ne sont plus hébergées sur le poste de l'utilisateur ou dans l'entreprise. Elles sont désormais dans le Cloud, accessibles par le réseau (généralement Internet) à l'aide d'un client léger (le navigateur, généralement). L'utilisateur peut donc désormais y accéder depuis n'importe où, avec différents supports (PC, smartphones, tablettes, etc).

2. Le modèle de consommation du logiciel : dans le modèle traditionnel, l'utilisateur achète une licence de l'application et en devient donc propriétaire, avec la possibilité de l'utiliser de manière illimitée. Les nouvelles versions impliquent l'achat d'une nouvelle licence. Dans le modèle SaaS, l'application peut être mise à disposition gratuitement (c'est le cas pour la majorité des webmails, comme GMail ou Hotmail), ou sur base d'un abonnement mensuel ou annuel, global ou lié au nombre d'utilisateurs (subscription-based pricing) et en fonction du niveau de services qu'il souhaite (par exemple, la quantité de Go qu'il réserve). Les mises à jour et la sécurité sont alors gérées par le fournisseur et sont transparentes pour l'utilisateur, qui ne doit plus effectuer d'actions spécifiques.

Exemples de fournisseurs SaaS : Google Apps, SalesForce, Microsoft Online Services, Oracle NetSuite, etc.

Ces différents modèles de services s'adressent à des profils d'utilisateurs différents, comme indiqué sur la Figure 2.3 : l'Infrastructure as a Service s'adresse à des gestionnaires de parcs informatiques, tandis que le Platform as a Service s'adresse à des intégrateurs, qui vont exploiter les fonctions mises à leur disposition pour créer de nouveaux logiciels. Le SaaS s'adresse quant à lui aux utilisateurs finaux, consommateurs d'applications.

La Figure 2.3 illustre également l'aspect pyramidal du Cloud Computing : le IaaS peut être utilisé pour fournir du PaaS, lui-même exploité pour délivrer du SaaS. L'utilisation du PaaS n'est toutefois pas obligatoire : un site Web pourrait directement s'implanter sur du IaaS et délivrer son service, sans faire de PaaS.

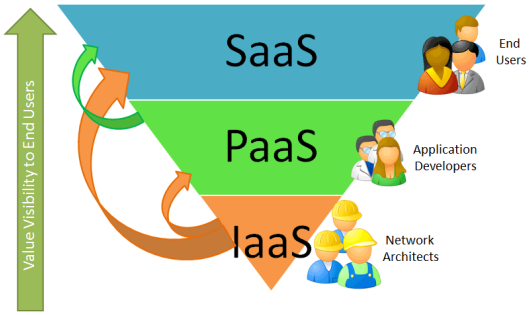


FIGURE 2.3 – Les modèles de services et leurs profils d'utilisateurs [50]

2.2 Acteurs

La mise en place d'une solution de type SaaS fait apparaître trois rôles différents : le SaaS User, le SaaS Provider et le Cloud Provider. Ces rôles vont se répartir les fonctions et responsabilités de la mise en oeuvre de la solution.

2.2.1 SaaS User

Le SaaS User est le consommateur final de l'application : nous le définissons ici comme un utilisateur ou une entreprise, qui exploite un service mis à disposition soit directement, soit en l'intégrant au sein de sa propre infrastructure informatique. Cette utilisation peut être gratuite ou non, sur base d'un abonnement mensuel ou annuel, en fonction du nombre d'utilisateurs ou global et du niveau de services souhaités. L'accès à ces ressources se fait par le réseau, généralement Internet, ce qui permet au client d'y accéder depuis n'importe où et depuis un grand nombre d'appareils différents.

2.2.2 SaaS Provider

Le SaaS Provider est fournisseur du SaaS User, à qui il va fournir une application et client du Cloud Provider, dont il loue les ressources informatiques. Son rôle est donc d'utiliser ces capacités informatiques mises à sa disposition pour lui permettre de délivrer aux SaaS Users un logiciel scalable, sécurisé, performant et hautement disponible. Il va donc prendre en charge le développement, le test, le déploiement et la maintenance de l'application qu'il met à disposition de ses clients.

2.2.3 Cloud Provider

Le Cloud Provider est l'entité responsable de la fourniture de ressources informatiques. Il a à sa disposition un ou plusieurs datacenters dont il est propriétaire ou locataire et dont il met les ressources à disposition du SaaS Provider, que nous venons de définir. Il est donc le gestionnaire du *Cloud*, tel que nous l'avons défini au chapitre 2. Les ressources mises à disposition sont de différents types : réseau, stockage, calcul, etc.

Les grandes sociétés actives dans le Cloud Computing, comme Amazon et Google, sont également clientes de leur propre infrastructure et se retrouvent donc avec les 3 rôles. Le meilleur exemple est Google avec sa messagerie Gmail : celle-ci se sert de l'infrastructure Cloud de Google pour proposer un service, que les employés de la firme eux-mêmes exploitent également.

La relation qui unit le SaaS Provider et le Cloud Provider se nomme l'Utility Computing.

2.2.4 Utility Computing

Pour définir l'Utility Computing, nous partirons du rapport [42], qui définit ce terme ainsi : *"l'Utility Computing est un ensemble de technologies et de pratiques commerciales qui permettent de délivrer une capacité de calcul de façon transparente et de manière fiable par l'utilisation de plusieurs ordinateurs. De plus, la capacité de calcul est disponible en fonction des besoins et facturée selon l'usage, un peu comme l'eau et l'électricité le sont aujourd'hui".*
En analysant cette définition, nous constatons que l'Utility Computing se compose dès le départ de deux aspects : un ensemble de technologies qui existent pour la plupart depuis des années et un business model - nommé "pay-per-use" ou "pay-for-what-you-use" - déjà utilisé dans d'autres secteurs, comme le gaz et l'électricité [10]. C'est très certainement cette caractéristique qui a fait réagir Larry Ellison ainsi : *"Ce qui est intéressant, avec le cloud computing, c'est que ce concept a été redéfini pour regrouper tout ce qui se fait déjà. Avec toutes ces annonces, absolument tout est Cloud Computing aujourd'hui".*

La définition reprise ci-dessus est toutefois incomplète, dans le sens où l'Utility Computing ne couvre pas uniquement la mise à disposition de capacités de calcul, même si celle-ci est sa finalité première. D'autres services vont également accompagner ce service de base, tels que :

- des ressources réseaux : bande passante, avec un volume et un débit définis, création de réseaux virtuels, etc.
- des ressources logicielles : licences, outils de gestion des ressources allouées, etc.
- des capacités de stockage.

Dans la suite de ce document, nous avons choisi de limiter volontairement

la définition du Cloud Computing à ce périmètre : le Cloud Computing sera la somme de l'Utility Computing et du SaaS, comme pour les auteurs de [6]. Dans ce contexte, les cloud privés et hybrides sont exclus de notre étude.

Cet ensemble de services s'accompagne de l'expertise nécessaire à leur gestion. Ainsi, l'Utility Computing est donc une forme de transfert de compétences : des compétences qui étaient précédemment gérées au sein de l'entreprise par l'équipe IT sont dorénavant "outsourcées" à l'extérieur et accessibles par le réseau, généralement Internet.

L'Utility Computing est donc la relation liant le fournisseur du Cloud et le fournisseur du SaaS : le premier met à disposition du second une infrastructure informatique, que ce dernier exploite pour fournir un service à ses propres utilisateurs, le plus souvent sous la forme d'une application Web.

La Figure 2.4 illustre le positionnement de l'Utility Computing, entre le fournisseur du Cloud et le fournisseur du SaaS [7].

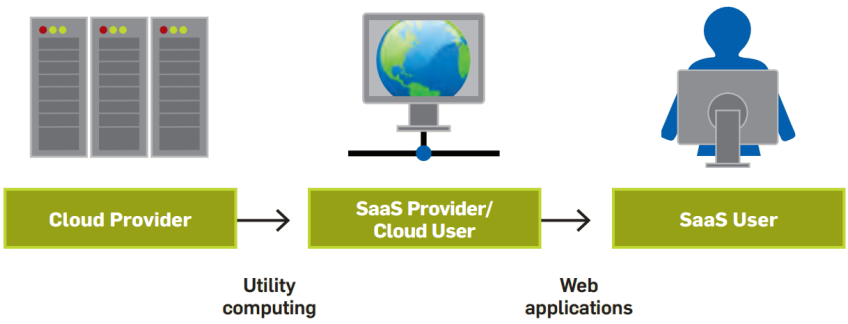


FIGURE 2.4 – L'Utility Computing [7]

Le business model lié à l'Utility Computing est, comme nous l'avons dit plus haut, inspiré des modèles existants pour le gaz, l'eau et l'électricité, c'est-à-dire que la facturation du service est basée sur la consommation par le client (*pay per use* en anglais). Le client, après avoir souscrit auprès d'un fournisseur, est donc facturé selon les ressources qu'il consomme. Dans ce cadre, des outils de mesure (*metering* en anglais) sont mis en place par le fournisseur pour permettre une quantification la plus précise possible des ressources utilisées. Ces mesures sont ensuite utilisées de différentes façons pour établir la facture : par exemple, les capacités de calcul et les licences sont facturées à l'heure d'utilisation, tandis que les données stockées sont facturées par giga-octets par mois.

Une seconde caractéristique de l'Utility Computing est liée à la quantité de ressources disponibles. Le fournisseur d'Utility Computing exploite généralement un ou plusieurs datacenters, dont les capacités sont mises à disposition des

clients. La taille de ces datacenters étant gigantesque (>1000 machines) et bien supérieure à ce qu'une seule entreprise peut se permettre de financer, le client a le sentiment de disposer d'une quantité infinie de ressources. Cette caractéristique, couplée au pay-per-use décrit plus haut, va avoir un impact sur la réalisation des traitements lourds : si par exemple le client souhaite faire une opération qui requiert 100 heures de temps CPU, il peut, moyennant un découpage en opérations parallèles, faire exécuter sa tâche sur 100 serveurs pendant 1 heure pour le même prix et obtenir ainsi un résultat plus rapidement.

La dernière caractéristique de l'Utility Computing est l'élasticité. L'élasticité est la capacité qu'offre le fournisseur au client d'augmenter ou réduire l'utilisation des ressources en fonction de ses besoins. Cet ajustement des ressources peut être soit manuel, soit adapté automatiquement à la charge requise.
La notion d'élasticité nous semble avoir toutefois une connotation trop commerciale : dans la suite de notre rapport, nous aborderons plutôt la notion de scalability, qui couvre des domaines et des compétences plus larges que l'augmentation de ressources matérielles.

Chapitre 3

Sécurité & vie privée

3.1 Sécurité

La sécurité se définit comme *"l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaire et mis en place pour conserver, rétablir, et garantir la sécurité du système d'information"* [57]. Les différents aspects de la sécurité que nous allons présenter ci-dessous se basent sur [32].

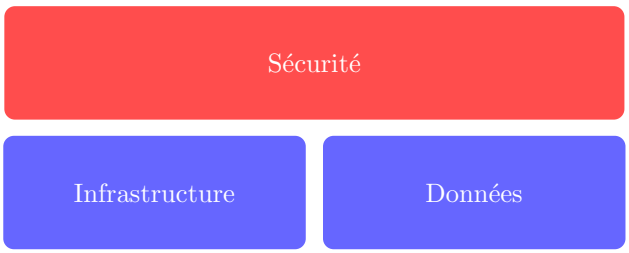


FIGURE 3.1 – Les différents thèmes de la sécurité

La Figure 3.1 propose un découpage des éléments de sécurité en 2 thèmes, que nous allons décrire et pour lesquels nous indiquerons quels aspects de la sécurité sont traités : l'infrastructure et les données.

3.1.1 Infrastructure

Nous définissons l'infrastructure ainsi : l'infrastructure représente les *"services qui rendent le Cloud et ses services disponibles aux utilisateurs finaux, ainsi que les mécanismes de transport du Cloud entre ses différents composants"* [27].

Du point de vue de l'utilisateur final, le SaaS Provider, de par l'implémentation de sa solution logicielle sur l'infrastructure du Cloud Provider, devient un de ces composants : la sécurisation de l'infrastructure va donc être partagée entre les deux acteurs, le Cloud Provider et le Saas Provider.

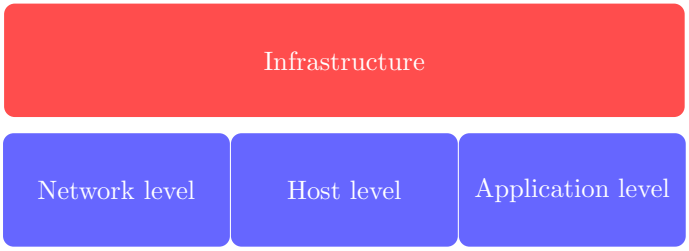


FIGURE 3.2 – L'infrastructure

Dans la suite de cette section, nous verrons la sécurité de l'infrastructure sous trois aspects : la sécurité de la couche réseau, la sécurité des hôtes et la sécurité au niveau applicatif, comme décrit à la Figure 3.2.

Au niveau réseau

La couche réseau va prendre en charge l'ensemble de la communication entre les différents composants du Cloud. Cette couche est particulièrement cruciale vu que l'ensemble des services du SaaS Provider et du Cloud Provider sont accessibles par un réseau, généralement Internet.

Les paramètres suivants vont relever de la couche réseau :

– La confidentialité : la confidentialité est une des bases de la sécurité de l'information et est *"le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé"* [55].

  Etant donné que les données de l'utilisateur final et que leur accès passent désormais par Internet, une stratégie de sécurité doit être mise en place pour assurer la confidentialité de celles-ci. Au niveau de la couche réseau, cet aspect va être traité par différents outils, comme les firewalls, la virtualisation du réseau, le cryptage et les tunnels sécurisés, qui seront répartis entre le Cloud Provider et le SaaS Provider.

Confidentialité	Cloud Provider	SaaS Provider
Firewall	X	
Network virtualization	X	
Encryption	X	X
Secured tunneling	X	X

– L'intégrité : l'intégrité est le fait d'assurer que l'information est correcte et consistante par rapport à l'état dans lequel elle est censée se trouver [36]. L'information n'a en effet de valeur que si elle est correcte [12]. Cette propriété *"s'applique aux données stockées, en transit ou en cours de traitement"* [13].

Au niveau de la couche réseau, l'intégrité des données va être assurée elle-aussi par du cryptage et des tunnels sécurisés. Le Cloud Provider aura également la possibilité d'utiliser des outils spécifiques de gestion de réseau, comme du route analytics.

Intégrité	Cloud Provider	SaaS Provider
Encryption	X	X
Secured tunneling	X	X
Route analytics	X	

– Le contrôle d'accès. Pour l'utilisateur final et le SaaS Provider, l'externalisation des ressources informatiques vers un Cloud a pour impact de perdre le contrôle sur l'audit du réseau. Celui-ci est en effet pris en charge intégralement par le Cloud Provider et il est très peu probable que le SaaS Provider ou l'utilisateur final y aient accès, ou aient la possibilité de collecter des données supplémentaires à ce niveau. La traçabilité du réseau relève donc de la compétence du Cloud Provider (network-level logs).

  Le partage d'une architecture Cloud entre différents consommateurs (les différents SaaS Providers hébergés) a également un impact : les attaques peuvent venir de l'extérieur (via Internet) ou de l'intérieur du Cloud. Des outils comme la virtualisation du réseau, des systèmes de détection d'intrusion et des règles de firewalls permettent de répondre à cette problématique.

Contrôle d'accès	Cloud Provider	SaaS Provider
Network-level logs	X	
Network Virtualization	X	
IDS	X	
Firewall	X	

– La disponibilité. La définition de la disponibilité (availability) sera précisée dans le chapitre 5, consacré à ce sujet.
  Comme tout système exposé sur Internet, l'infrastructure Cloud doit être protégée des problèmes "classiques" : mauvaise configuration de DNS, attaques de DNS, DNS cache poisoning, denial of service (DoS) et distributed denial of service (DDoS). Ces dernières attaques, le DoS et le DDoS, peuvent également survenir depuis "l'intérieur" du Cloud : les ressources réseaux étant partagées, un autre client du Cloud pourrait attaquer votre système depuis le réseau interne.

  Les problèmes de disponibilité du Cloud sont donc gérés par des outils habituels, c'est-à-dire les firewalls et les systèmes de détection d'intrusion (IDS). La problématique des attaques internes sera gérée également par les firewalls et la virtualisation du réseau.

Disponibilité	Cloud Provider	SaaS Provider
Network Virtualization	X	
IDS + IPS	X	
Firewall	X	

Au niveau de l'hôte

L'hôte représente ici la ressource informatique de calcul mise à disposition du SaaS Provider par le Cloud Provider. Cette ressource peut prendre deux formes :

1. Un environnement de développement, déploiement et hébergement virtuel : il s'agit ici d'une configuration de type PaaS (Platform as a Service). Dans cette configuration, l'aspect sécurité de l'hôte est opaque et pris en charge par le Cloud Provider : le système d'exploitation est masqué et une couche d'abstraction permet au SaaS Provider d'interagir avec lui, dans des limites clairement établies.

2. Une machine virtuelle : c'est le cas avec une configuration de type IaaS (Infrastructure as a Service). Dans ce cas, le Cloud Provider met à disposition du SaaS Provider une configuration de base, avec un système d'exploitation et une sécurité par défaut, que le SaaS Provider peut adapter entièrement en fonction de ses besoins. La sécurité va donc se situer à deux niveaux :
   - Au niveau du logiciel de virtualisation : le logiciel de virtualisation, généralement nommé hypervisor et présenté à la Figure 3.3, est la couche logicielle entre le matériel et les machines virtuelles qui permet au SaaS Provider de créer et gérer ses instances. Ce logiciel et ses aspects sécuritaires sont entièrement gérés le Cloud Provider.
     La sécurisation de l'hypervisor est une des priorités du Cloud Provider. Les techniques utilisées pour cela sont relativement classiques : monitoring des events, utiliser un environnement de tests, limiter les accès et la surface d'attaque [47].

Hypervisor	Cloud Provider	SaaS Provider
Events monitoring	X	
Limit access	X	
Limit threat surface	X	

   - Au niveau des instances virtuelles : les instances virtuelles sont créées et gérées par le SaaS Provider, à partir d'images qui incluent un système d'exploitation. La sécurisation de cette instance relève alors de sa responsabilité : les techniques habituelles de protection d'une machine connectée à Internet sont appliquées (firewall, antivirus, etc).

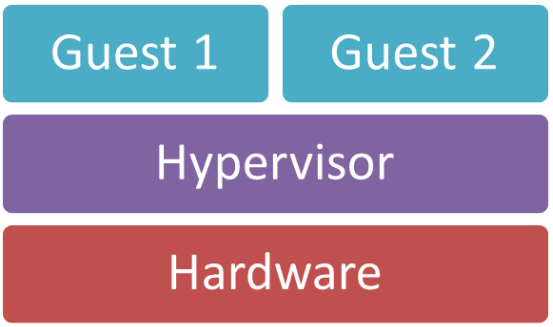


FIGURE 3.3 – L'hypervisor [11]

Instances	Cloud Provider	SaaS Provider
Events monitoring		X
Limit access		X
Limit threat surface		X
Antivirus		X
Firewall		X
IDS + IPS		X
Strong authentication		X

Au niveau applicatif

La sécurité au niveau de l'application va porter sur trois thèmes :

1. L'isolation : nous définissons ici l'isolation comme la séparation existant entre les différents environnements de consommateurs, qui les empêchent de rentrer en conflit les uns avec les autres.

   L'isolation entre les différents services offerts à l'utilisateur final va se répartir entre le Cloud Provider et le SaaS Provider en fonction de l'implémentation des fonctionnalités chez l'un ou chez l'autre. Par exemple, le multitenancy (cfr 4) peut être géré du côté Cloud ou du côté SaaS. Parmi les autres outils relatifs à l'isolation, nous trouverons le monitoring et l'architecture sandbox.
   L'outil principal utilisé pour l'isolation est la virtualisation : avec celle-ci, chaque SaaS Provider bénéficie de son propre environnement, distinct des autres SaaS Providers.

Isolation	Cloud Provider	SaaS Provider
Sandbox architecture	X	
Monitoring	X	X
Multitenancy	X	X
Virtualization	X	

2. L'authentification : l'authentification est *le processus qui détermine si quelqu'un ou quelque chose est en réalité celui ou ce qu'il déclare être* [43].

   L'authentification des utilisateurs finaux, qui vont pouvoir consommer le service proposé par le SaaS Provider, va être soit prise en charge par le SaaS Provider, soit par le Cloud Provider. Ce dernier cas se produira généralement lorsque la relation liant le SaaS Provider et le Cloud Provider est du PaaS (Platform as a Service).

Authentification	Cloud Provider	SaaS Provider
Monitoring	X	X
User authentication	X	X
Single sign-on	X	X
SSL or TLS	X	X

3. Le contrôle d'accès : à nouveau, la prise en charge du contrôle d'accès va se répartir entre le SaaS Provider et le Cloud Provider, en fonction de leur type de relation (IaaS ou PaaS). Les outils seront la gestion des privilèges et rôles, ainsi que le monitoring.

Contrôle d'accès	Cloud Provider	SaaS Provider
Monitoring	X	X
Privilege management	X	X

3.1.2 Données

L'aspect transfert des données ayant été traité dans le paragraphe 3.1.1 de la section 3.1.1 "infrastructure", nous aborderons ici l'aspect stockage (data-at-rest en anglais). Le stockage des données est principalement du ressort du Cloud Provider, qui va bâtir une infrastructure physique conséquente pour assurer ce rôle. Le SaaS Provider, en tant que tel, n'a pas un rôle de stockage : il agit uniquement comme un intermédiaire entre l'utilisateur final (SaaS User) et le Cloud Provider.

Dans leur livre [32], les auteurs identifient trois thèmes liés au stockage de données dans le Cloud :

1. La confidentialité : la confidentialité consiste à rendre *"l'information accessible uniquement à ceux dont l'accès est autorisé"* [13]. Au niveau des données, cela va se faire au moyen de deux outils :

   – Le contrôle d'accès à la donnée : cette fonctionnalité est généralement délivrée par le Cloud Provider, mais dans une version très limitée. Par

exemple, le Cloud Provider va définir deux grands rôles (administrateur et utilisateur), sans possibilité d'ajouter des rôles supplémentaires. Cet outil risque donc de ne pas convenir à des grandes entreprises, où la confidentialité des documents requiert différents niveaux d'accès. L'extension de cet outil peut être prise en charge par le SaaS Provider : celui-ci propose alors à l'utilisateur final un mécanisme plus poussé de contrôle d'accès.

– La protection de la donnée : la protection des données va passer par le cryptage. Celui-ci peut être proposé nativement par le Cloud Provider, mais il peut également être implémenté par le SaaS Provider si nécessaire. Un élément important est toutefois à noter : le cryptage d'une donnée interdit son indexation ultérieure, ce qui explique que généralement, les données hébergées par un SaaS ne sont pas cryptées.

Confidentialité	Cloud Provider	SaaS Provider
Contrôle d'accès	X	X
Encryption	X	X

2. L'intégrité : le cryptage des données assure la confidentialité, mais pas leur intégrité. L'intégrité peut être prise en charge par des mécanismes de type MAC (Message Authentication Code), qui agissent comme un mécanisme de hashing et qui peuvent être proposés par le Cloud Provider ou le SaaS Provider. Le SaaS Provider a également la possibilité de faire des contrôles par programmation [31].

Intégrité	Cloud Provider	SaaS Provider
MAC	X	X
Programmatic checks		X

3. La disponibilité : la disponibilité des données va dépendre de trois éléments :
   (a) La stratégie de disponibilité au niveau du réseau (cfr 3.1.1) que le Cloud Provider aura mis en place ;
   (b) La disponibilité intrinsèque du Cloud Provider, exprimée en pourcentage d'uptime ;
   (c) La pérénnité du Cloud Provider.

   La disponibilité des données peut être assurée par les outils habituels, qui peuvent être mis à disposition soit par le Cloud Provider, soit par le SaaS Provider : backup, redondance, etc.

Disponibilité	Cloud Provider	SaaS Provider
Backup	X	X
Redondance	X	X

3.2 Vie privée

Définir la vie privée (privacy) est complexe, étant donné que ce concept varie en fonction du pays, de la culture et de la juridiction en cours [32]. Cette diversité implique une absence de standard universel sur la vie privée. Nous utiliserons donc la définition relativement générique de [35] : la vie privée représente *"les droits et obligations des individus et des entreprises en ce qui concerne la collecte, l'utilisation, la conservation, la divulgation et l'élimination d'informations personnelles"*. Le terme généralement utilisé en français est "vie privée".

Cette définition nous amène à préciser également la notion d'"informations personnelles", qui n'est pas non plus un terme universellement défini. Dans notre cas, nous poserons la définition suivante : une information personnelle est *"toute information qui porte sur ou qui peut être liée à un individu identifié ou identifiable"* [35].

Dans un système informatique traditionnel, la responsabilité revient à l'entreprise : cela couvre la collecte, l'utilisation, le stockage, la protection et la destruction. L'émergence du Cloud Computing change la donne, car les données ne sont plus stockées au sein de l'entreprise et sont gérées par le Cloud Provider [34]. Ce transfert de compétences implique un partage de responsabilités et d'obligations entre le SaaS Provider qui va collecter les données et les exploiter, et le Cloud Provider qui va assurer leur stockage, leur protection et leur destruction éventuelle.
L'absence de règlementation internationale à ce niveau ajoute une complexité : les données étant hébergées dans le cloud, l'utilisateur final a peu de visibilité sur leur localisation effective et donc sur la juridiction qui leur est appliquée.

Le Cloud Computing a donc un impact important sur la vie privée. Il n'existe toutefois pas d'outils spécifiques permettant à une application de respecter la vie privée, hormis ceux que nous avons cités pour la sécurité dans le début de ce chapitre. Il existe par contre une série de principes communément admis sur la vie privée qui servent de bonnes pratiques. Les auteurs de [32] en indiquent 6 qui sont impactés par le Cloud Computing :

1. La limitation de la collecte : ce principe spécifie que le minimum d'informations personnelles doit être collecté, et ce, avec le consentement et la connaissance du sujet. Ce principe s'applique exclusivement au SaaS Provider, qui va déterminer dans son application les informations qui sont strictement nécessaires à son fonctionnement.

2. La limitation de l'usage : l'usage, la divulgation et la mise à disposition des informations personnelles doivent être strictement limités aux rôles auxquels la personne concernée les a consentis ou à ce que la législation impose. Le SaaS Provider et le Cloud Provider ne peuvent donc utiliser les données à d'autres fins que celles accordées par leur propriétaire.

3. La sécurité : ce principe impose de protéger les données confidentielles contre la perte, le vol, l'accès non autorisé, la destruction, l'usage, la modification ou la divulgation. Les différents outils abordés dans la section sécurité de ce chapitre en font donc partie et concernent aussi bien le Cloud Provider que le SaaS Provider.

4. La rétention et la destruction : les données personnelles ne doivent pas être conservées plus longtemps que strictement nécessaire. Une fois le délai expiré, elles doivent être détruites. La durée de la rétention est donc clairement de la responsabilité du SaaS Provider, qui est le seul à même de déterminer leur durée de vie. Etant donné que le stockage est fourni par le Cloud Provider, la destruction relève de sa compétence mais plusieurs difficultés apparaissent à ce niveau :
   - la réplication et les backups doivent également supprimer les données personnelles ;
   - les espaces disques occupés par des fichiers devraient être réécrits plusieurs fois pour ne plus permettre leur récupération (7x, d'après le gouvernement fédéral des USA) ;
   - la virtualisation et la réallocation du stockage impliquent que d'autres utilisateurs pourraient avoir accès à des données supprimées ;
   - le matériel remplacé doit être systématiquement détruit physiquement.

   Une solution pour garantir la suppression des données serait d'utiliser du cryptage : si les données doivent être supprimées, il suffit de supprimer la clé d'encryption pour ne plus être en mesure d'y accéder.

5. Le transfert : ce principe indique que les données ne devraient pas être transférées vers un autre pays qui n'applique pas le même niveau de protection de la vie privée que ce que l'entreprise propose. Cette responsabilité incombe donc au SaaS Provider et au Cloud Provider : le Cloud Provider doit permettre de limiter la localisation des données à certains pays compatibles avec le niveau de protection requis et le SaaS Provider doit s'assurer que les différents services qu'il pourrait lui-même exploiter atteignent également ce niveau de protection.

6. La responsabilité : l'organisation qui a le contrôle des données en est responsable et doit désigner une ou plusieurs personnes qui sont responsables de la compliance de l'entreprise vis-à-vis des autres principes [32]. Une solution serait de fixer une politique d'utilisation des données personnelles et des mécanismes de contrôle du respect de cette politique par tous les acteurs en relation avec ces données.

La prise en charge des principes de la vie privée se répartit donc ainsi :

Privacy	Cloud Provider	SaaS Provider
Collection limitation		X
Use limitation	X	X
Security	X	X
Retention		X
Destruction	X	
Destruction via encryption		X
Transfer	X	X
Accountability	X	X

3.3 Synthèse de la sécurité et de la vie privée

Le tableau suivant reprend l'ensemble des caractéristiques et outils liés à la sécurité d'une solution Cloud. La symbolique utilisée est la suivante :

- C = c'est au Cloud Provider à implémenter l'outil
- S = c'est au SaaS Provider à implémenter l'outil
- C & S = l'outil doit être implémenté par le Cloud Provider et SaaS Provider
- C|S = l'outil doit être implémenté par le Cloud Provider, le SaaS Provider ou les 2
- C/S = l'outil doit être implémenté par Cloud Provider ou (exclusif) le SaaS Provider

TABLE 3.1: Tableau synthétique sur la répartition des outils de sécurité et de vie privée dans une solution Cloud

Sécurité			Outils				
	Infrastructure						
		Au niveau réseau					
				Confidentialité	Intégrité	Contrôle d'accès	Disponibilité
			Firewall	C		C	C
			Network virtualization	C		C	C
			Encryption	C/S	C/S		
			Tunneling	C/S	C/S		
			Route analytics		C		
			Network-level logs			C	
			IDS			C	C
		Au niveau hôte (IaaS)					
			Hypervisor	Confidentialité	Intégrité	Contrôle d'accès	Disponibilité
			Events monitoring			C	
			Limit access	C	C	C	C
			Limit threat surface	C	C	C	C
			Instances	Confidentialité	Intégrité	Contrôle d'accès	Disponibilité
			Events monitoring			S	
			Limit access	S	S	S	S
			Limit threat surface	S	S	S	S
			Antivirus	S	S		
			Firewall	S		S	
			IDS/IPS	S		S	
			Strong authentication	S		S	

		Au niveau applicatif					
				Authentification	Contrôle d'accès	Isolation	
			Sandbox architecture			C	
			Multitenancy			C|S	
			Virtualization			C	
			Monitoring	C|S	C|S	C|S	
			User authentication	C|S			
			Single sign-on	C|S			
			Privilege management		C|S		
			SSL or TLS	C/S			
			Secured APIs	C	C		
	Données						
				Confidentialité	Intégrité	Disponibilité	
			Contrôle d'accès	C|S			
			Encryption	C/S			
			MAC		C/S		
			Programmatic checks		S		
			Backup			C/S	
			Redondance			C/S	

Vie privée			Outils				
			Collection limitation	S			
			Use limitation	C & S			
			Security	C & S			
			Retention	S			
			Destruction	C			
			Destruction via encryption	S			
			Transfer	C & S			
			Accountability	C & S			

Chapitre 4

Scalability

Comme l'a écrit l'auteur de [23], la scalability n'a pas une définition scientifique rigoureuse et avalisée par l'ensemble de la communauté. L'auteur suggère même d'abandonner l'utilisation de ce terme. Malgré tout, cette caractéristique est régulièrement citée lorsque le Cloud Computing est abordé, c'est pourquoi nous allons utiliser ce terme et nous attacher à le définir le plus précisément possible.

La scalability (évolutivité en français) peut se définir ainsi : il s'agit de *la capacité d'un dispositif informatique à s'adapter au rythme de la demande* [40]. Dans le cas de notre étude technologique sur le Software as a Service, nous devons affiner cette définition : il s'agit de la capacité du fournisseur du SaaS à pourvoir aux besoins de ses clients par la mise à disposition des ressources informatiques suffisantes au traitement des requêtes transmises. Il s'agit là du véritable défi technique et technologique du fournisseur de SaaS.

Toutefois, avant de décrire ces différentes sections, il convient de définir un outil extrêmement important utilisé pour la scalability du stockage et de l'application du SaaS Provider : le multi-tenant et le single-tenant.

Multi-tenant / Single-tenant

Les caractéristiques multi-tenancy et single-tenancy sont des patterns d'architecture logicielle opposés qui vont être sélectionnés lors de la mise en place de la solution Cloud. Le pattern multi-tenant est généralement implémenté par le Cloud Provider, mais rien n'interdit que le SaaS Provider l'implémente lui-même.

Un tenant se définit ainsi : il s'agit d'une *d'une entité organisationnelle qui loue une application d'un SaaS provider. Un tenant regroupe donc un ensemble d'utilisateurs.*[9]. Dans le cas de l'architecture multi-tenant, les clients du SaaS Provider, appelés les tenants, se connectent à une seule et même instance du logiciel [59].

Cette forme d'architecture implique un niveau d'abstraction supplémentaire

lors de la conception : étant donné que les tenants vont partager la même instance du logiciel et en particulier, la même instance de base de données, une couche supplémentaire devra être mise en place pour permettre à tous les tenants d'adapter l'application à leurs besoins spécifiques en terme de processus métier et d'ergonomie [16]. Cette abstraction supplémentaire est généralement appelée méta-modélisation [45].

Il s'agit là d'une différence par rapport à un environnement multi-utilisateur : l'application doit être conçue dès le départ pour permettre un cloisonnement stricte entre les données, configurations et flux des différents tenants, sans interférence entre eux, alors que dans une configuration multi-utilisateurs, les processus métiers de base et les flux sont communs à tous les utilisateurs de l'application. Il faut donc considérer une architecture multi-tenant comme une encapsulation d'un système multi-utilisateurs, comme représenté à la Figure 4.1.

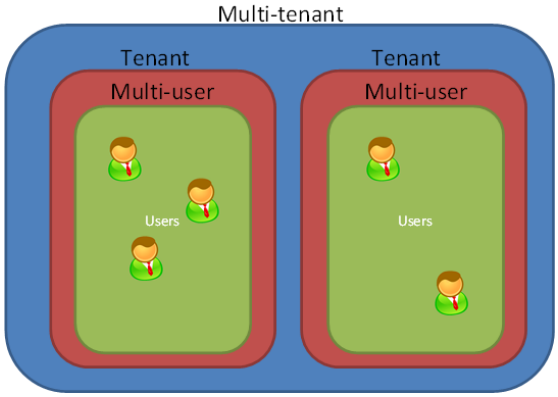


FIGURE 4.1 – Multi-tenant architecture

La Figure 4.1 illustre un exemple d'architecture multi-tenant, dans laquelle chaque tenant dispose d'un système multi-utilisateurs.

Une architecture multi-tenant apporte plusieurs avantages [9] :

- L'optimisation de l'utilisation des ressources matérielles : les mêmes ressources matérielles étant partagées entre les différents tenants, le SaaS provider est en mesure d'augmenter ou diminuer la puissance disponible de façon globale, pour l'ensemble de ses clients.

- La facilité de la maintenance : une instance de l'application et une de la base de données doivent être maintenues, ce qui limite les opérations à réaliser ;

- La création de nouvelles agrégations de données : la centralisation des données de l'ensemble des utilisateurs dans une seule instance va permettre au SaaS provider d'étudier de nouvelles agrégations possibles des données, en vue d'optimiser l'utilisation de la base de données.

Ces différents avantages vont permettre une réduction générale du coût de la solution pour le Cloud Provider et/ou le SaaS Provider et donc augmenter leur compétitivité par rapport à la concurrence.

Salesforce.com est un exemple typique de SaaS provider ayant implémenté une architecture multi-tenant [45].

Une architecture single-tenant va passer par la création d'un environnement spécifique à chaque tenant. Chaque tenant possède donc sa propre instance de l'application et de la base de données, avec éventuellement un matériel, généralement virtualisé, spécifiquement attribué.

A partir de la section suivante, nous allons analyser la scalability et la répartition de sa prise en charge entre le SaaS Provider et le Cloud Provider. Nous indiquerons ensuite quelle partie peut implémenter le caractère multi-tenant ou single-tenant.

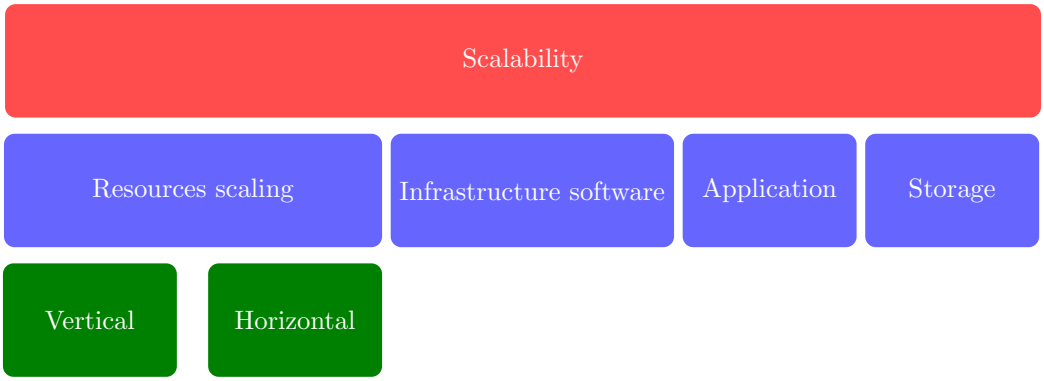


FIGURE 4.2 – Les différents thèmes de la scalability

La Figure 4.2 ci-dessus présente la structure que nous allons utiliser pour décrire les fonctionnalités et outils de la scalability.

4.1 Echelonnement des ressources

L'ajout ou le retrait des ressources pour répondre aux besoins de l'utilisateur final va clairement dépendre du Cloud Provider : celui-ci a en effet la main sur l'attribution des ressources virtuelles et donc des ressources réelles qui seront exploitées.

L'attribution de ces ressources peut se faire de deux façons : par du scaling vertical (scale-up) ou du scaling horizontal (scale-out) [44]. Dans cette section,

nous limiterons volontairement l'échelonnement des ressources (resources scaling) à la mise à disposition de ressources supplémentaires. La section suivante, architecture, abordera quant à elle les aspects logiciels nécessaires à la scalability.

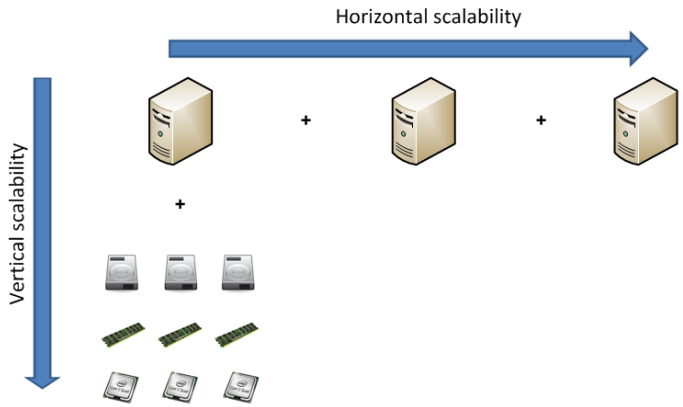


FIGURE 4.3 – Le scaling horizontal & vertical

La Figure 4.3 nous indique les différences entre ces deux types de scaling :

- Le scaling vertical se définit par *"la possibilité d'augmenter (ou diminuer) les capacités d'un service Cloud comme une machine virtuelle en augmentant ses ressources comme la mémoire physique, la vitesse du CPU ou la bande passante"* [19]. Ce scaling va se traduire pour le Cloud Provider par la possibilité d'augmenter les ressources d'une instance virtuelle via du hot-swap/cold-swap ou par la mise à disposition de différentes configurations d'instances, comme Amazon le propose dans son service EC2 [1]. Celui-ci permet d'instancier aisément différentes configurations types, allant des micro-instances à des instances extra-larges, avec des orientations définies : usage général, calcul optimisé, mémoire optimisée, stockage optimisé, GPU.

- Le scaling horizontal représente quant à lui *"l'augmentation (ou la diminution) des ressources Cloud de mêmes types, comme initialiser davantage de machines virtuelles du même type durant un pic de charge"* [19]. Cette augmentation d'instances implique que l'architecture logicielle soit capable d'exploiter ces nouvelles ressources mises à disposition, ce qui relève du SaaS Provider. Nous décrirons les aspects nécessaires de l'architecture logicielle dans la section suivante. Pour le Cloud Provider, il va s'agir de permettre au SaaS Provider d'instancier manuellement ou automatiquement de nouvelles ressources.

Tool	Vertical	Horizontal	Cloud Provider	SaaS Provider
Add/remove instance ressources	X		X	
Add/remove new instances		X	X	
Hot-swapping/Cold-swapping	X		X	
Multiple configurations	X		X	

4.2 Logiciel d'infrastructure

Le logiciel d'infrastructure se présente comme la couche logicielle fournie par le Cloud Provider permettant de gérer l'allocation et la désallocation des ressources en terme de capacité de calcul, de réseau et de stockage. Cette couche va utiliser les fonctionnalités suivantes pour être "scalable" :

– Le load-balancing : conceptuellement, il s'agit *"d'un pont entre les serveurs et le réseau"* [28], qui a pour but de *"distribuer la charge entre différents serveurs pour aller au-delà de la capacité d'un seul et autoriser la panne de l'un de ceux-ci."*[28], comme indiqué dans la Figure 4.4. Le load-balancing peut concerner les serveurs, les firewalls ou les caches. Le Cloud Provider dispose généralement de solutions de load-balancing, comme Amazon avec son service Elastic Load Balancing [3], mais des solutions logicielles existent (comme Resonate, Rainfinity et Stonebeat [28]) et pourraient être implémentées par le SaaS Provider.

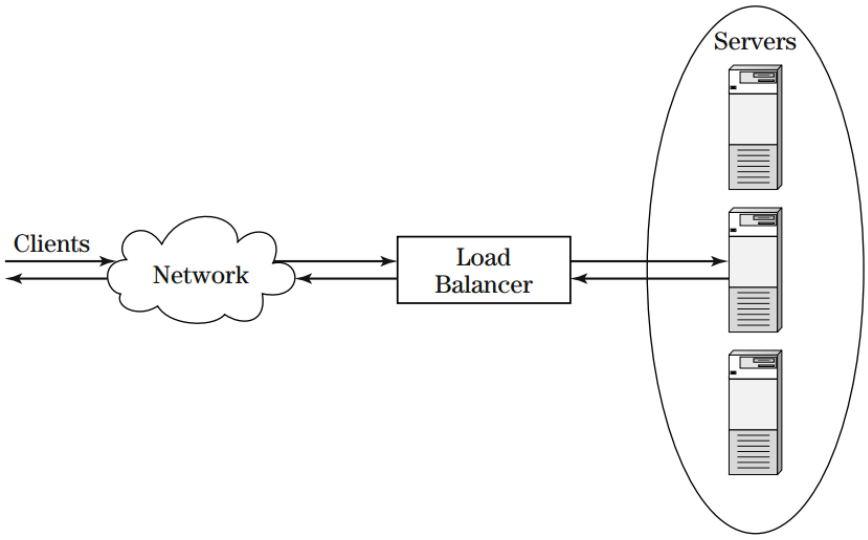


FIGURE 4.4 – Un load-balancer devant un ensemble de serveurs [28]

– L'auto-scaling : l'auto-scaling est une fonctionnalité implémentée par le Cloud Provider qui permet d'automatiquement *augmenter ou diminuer*

les capacités des machines virtuelles en se basant sur des règles[30]. Les règles consistent en des métriques de performances et des seuils que l'utilisateur (dans notre cas, le SaaS Provider) adapte. AWS Auto Scaling [2] est un exemple de ce type de service.

– Les snapshots : un snapshot est *"l'état d'un système à un moment particulier"* [61]. Dans le cas qui nous intéresse, l'utilisation de snapshots va permettre de créer une image de base qui sera utilisée pour créer très rapidement une série d'instances de machines virtuelles.

En résumé de la partie logiciel d'infrastructure, nous avons les outils suivants :

Infrastructure software	Cloud Provider	SaaS Provider
Load balancing	X	X
Auto-scaling layer	X	
Snapshots	X	

4.3 Application

L'application est le logiciel fourni par le SaaS Provider qui va exploiter les ressources du Cloud Provider pour délivrer un ensemble de services à l'utilisateur final au moyen des outils suivants :

– Le couplage faible : pour obtenir une solution "scalable", l'architecture, et en particulier l'application, doit être elle-même "scalable", c'est-à-dire qu'elle doit être constituée *d'un ensemble de composants qui peuvent évoluer de façon indépendante les uns par rapport aux autres* [26]. Cette approche, appelée également SOA pour Service Oriented Architecture, assure un couplage faible entre chaque élément et va permettre de dimensionner les ressources nécessaires à chacun d'entre eux, en fonction de leurs besoins. La Figure 4.5 illustre les différents éléments d'une architecture exemple, pour lesquels un certain nombre de ressources a été alloué, indépendamment les uns des autres.

– Le software tuning : le software tuning est *"l'adaptation du logiciel à un ensemble de conditions de calcul."* [51]. Dans notre cas, nous devons préciser cette définition : il s'agit d'optimiser les performances du logiciel en fonction des ressources qui lui sont allouées. Par exemple, le logiciel doit être capable d'exploiter le plus efficacement possible la mise à disposition de mémoire supplémentaire. Cet outil est du ressort du SaaS Provider et s'applique à la partie "application" de la solution.

– Le multitenancy, qui peut être implémenté par le Cloud Provider (principalement lorsque celui-ci fournit du PaaS) ou par le SaaS Provider.

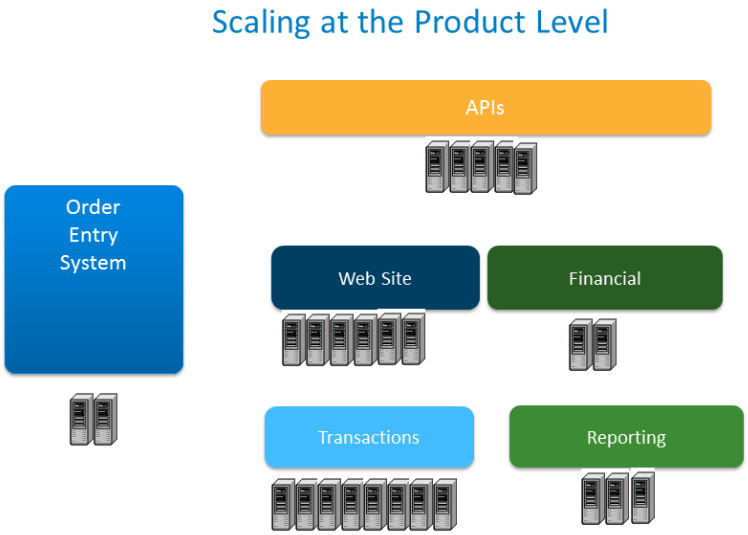
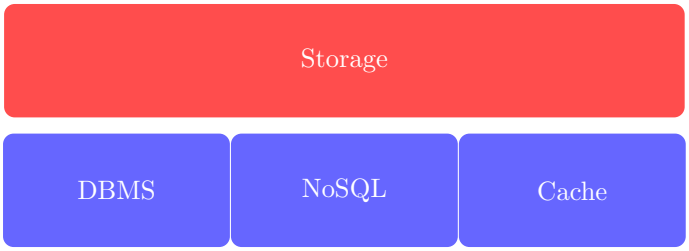


FIGURE 4.5 – L'augmentation des ressources au niveau des composants indépendants [26]

En résumé de la partie application, nous avons les outils suivants :

Application	Cloud Provider	SaaS Provider
SOA/Minimal coupling		X
Application tuning		X
Multitenancy	X	X

4.4 Stockage



Dans leur article *"Dynamically scaling applications in the cloud"* [53], les auteurs relèvent trois mécanismes généralement proposées par les Cloud Provider

pour assurer le stockage des données :

1. Les bases de données traditionnelles (DBMS) : certains Cloud Providers, comme Amazon avec RDS et Microsoft avec SQL Server, proposent dans leurs offres des bases de données traditionnelles, basées sur SQL et les propriétés ACID (atomicité, cohérence, isolation et durabilité). La scalability de ce type de base de données passe généralement par l'utilisation de partioning et du clustering :

   – Le partitioning a pour objectif de répartir les données entre plusieurs noeuds sur base d'une caractéristique des données (la valeur d'un attribut, la date d'insertion, etc), ce qui permet de stocker une grande quantité de données en les répartissant sur plusieurs machines et plusieurs volumes de stockage ;

   – Le clustering réplique les données sur plusieurs noeuds dans le but d'augmenter les performances par une répartition de la charge.

   Les base de données traditionnelles sont toutefois peu adaptées à l'architecture Cloud : l'utilisation de transactions, qui permet d'assurer la cohérence des données, nécessite de protéger la donnée des autres demandes d'accès durant toute la durée de la transaction, ce qui rend la donnée inaccessible [53]. Donc, plus les transactions sont nombreuses, plus les conflits apparaissent et compromettent la performance apportée par le clustering. Ce type de base de données est donc davantage adapté au scaling vertical (augmentation de ressources de l'instance) qu'au scaling horizontal (augmentation du nombre d'instances). Pour limiter ces difficultés, certains éditeurs de base de données travaillent à l'amélioration de la réplication des données. Une autre solution est de passer par un middleware qui agit en front-end des bases de données et transforme les opérations pour assurer que toutes les copies des données sont à jour [53]. DBFarm [39] est un exemple de ce type de middleware.

2. Les bases de données NoSQL : ce terme fait référence à une grande variété de systèmes et de fonctionnalités différents, dont la caractéristique principale est l'implémentation des propriétés BASE : Basic Availability Soft-state Eventual consistency. Là où une base de données traditionnelles implémente ACID et met l'accent sur la consistance par l'utilisation de transactions, une base de données NoSQL est orientée prioritairement sur la disponibilité. Elle possède en outre les caractéristiques suivantes [14] :
   – non-relationnelle
   – distribuée
   – horizontalement scalable
   – schema-free
   – prévue pour de grandes quantités de données

   Ce type de base de données est donc tout à fait en mesure d'exploiter l'architecture d'un Cloud pour offrir un maximum de scalability. Toute-

fois, les mécanismes de gestion de la réplication offre moins de garanties que les systèmes traditionnels. Par exemple, il se peut que la mise à jour d'une donnée ne soit pas effective immédiatement et qu'elle soit postposée à un moment ultérieur [53]. T. Perdue, dans son article *"NoSQL : An Overview of NoSQL Databases"* [37], relève quatre catégories de base de données NoSQL :

(a) Les Key-values Stores : dans cette catégorie, chaque donnée est identifiée par un identifiant unique qui permet de la récupérer et de la mettre à jour. Le service Dynamo d'Amazon Web Services est un exemple de Key-values Store.

(b) Les Column Family Stores : cette catégorie est très similaire aux Key-values Stores. La clé référence ici une ligne contenant différentes colonnes. Les logiciels de cette catégorie les plus courants sont Big-Table de Google et Cassandra.

(c) Les Document Databases : il s'agit à nouveau ici d'un modèle similaire au Key-Values mais implémentant également le versionning des données.

(d) Les Graph Databases : dans ce type de base de données NoSQL, les éléments sont des noeuds, ayant des propriétés et des relations entre eux. Neo4J est un exemple de ce type.

3. L'utilisation d'un cache distribué : les systèmes de cache sont *utilisés pour stocker les données intermédiaires en vue d'accélérer les requêtes d'accès à ces données* [53]. Ils fonctionnent en complément d'une base de données traditionnelle ou NoSQL et permettent de limiter les accès à celle-ci, diminuant de facto la charge. Microsoft et Amazon proposent tous les deux cette fonction :le service Caching pour Windows Azure et ElastiCache pour AWS.

Le tableau 4.1 [6] nous résume les différents modèles de stockage proposés par Amazon, Microsoft et Google.

Stockage	Cloud Provider	SaaS Provider
SQL-based databases	X	X
NoSQL databases	X	X
Caching services	X	X
Multitenancy	X	X

Amazon Web Services	Microsoft Azure	Google AppEngine
– Range of models from block store (EBS) to augmented key/blob store (SimpleDB) – Automatic scaling varies from no scaling or sharing (EBS) to fully automatic (SimpleDB, S3), depending on which model used – Consistency guarantees vary widely depending on which model used – APIs vary from standardized (EBS) to proprietary	– SQL Data Services (restricted view of SQL Server) – Azure storage service	– MegaStore/BigTable

TABLE 4.1 – Les différents modèles de stockage d'Amazon Web Services, Microsoft Azure et Google AppEngine [6]

4.5 Synthèse de la scalability

Le tableau suivant reprend l'ensemble des caractéristiques et outils liés à la scalability d'une solution Cloud. La symbolique utilisée est identique à celle utilisée dans le chapitre 3 "Sécurité".

TABLE 4.2 – Tableau synthétique sur la répartition des outils de scalability dans une solution Cloud

Resources scaling		Outils	
	Vertical scaling		
		Add/remove instance resources	C
		Hot-swapping/Cold-swapping	C
		Multiple configurations	C
	Horizontal scaling		
		Add/remove new instances	C
Infrastructure software			
		Load balancing	C/S
		Auto-scaling layer	C
		Snapshots	C
Application			
		SOA / Minimal coupling	S
		Application tuning	S
		Multitenancy	C/S
Stockage			
		SQL-based databases	C/S
		NoSQL databases	C/S
		Caching services	C/S
		Multitenancy	C/S

Chapitre 5

Disponibilité

Dans leur article [46], les auteurs indiquent que la disponibilité signifie *"qu'un système est en ligne et prêt à répondre"*. La disponibilité a donc pour objectif de minimiser le temps d'arrêt (downtime) d'un service et de minimiser le temps nécessaire pour restaurer la situation après une panne. Comme l'indiquent les auteurs de [46], la disponibilité n'est pas une technologie spécifique, il s'agit d'un objectif à atteindre qui va impacter le coût, la compréhensibilité et la complexité de la solution.

Une métrique largement répandue pour mesurer la disponibilité d'une solution est basée sur la formule $\frac{MTBF}{MTBF+MTTR}$, où MTBF représente le temps moyen entre les pannes (Mean Time Between Failures) et MTTR le temps moyen de réparation (Mean Time To Repair). Le coefficient ainsi obtenu permet une classification de la disponibilité d'un système, comme indiqué dans la Table 5.1.

System Type	(min/year)	Availability	Class
unmanaged	50,000	90.%	1
managed	5,000	99.%	2
well-managed	500	99.9%	3
fault-tolerant	50	99.99%	4
high-availability	5	99.999%	5
very-high-availability	.5	99.9999%	6
ultra-availability	.05	99.99999%	7

Table 5.1 – Les différentes classes de disponibilité [20]

Dans la collaboration qui unit le Cloud Provider et le SaaS Provider, le coefficient de disponibilité sera le minimum des coefficients des deux parties. Donc, si le SaaS Provider fournit 99,999% de disponibilité et que le Cloud Provider est limité à 99,95%, le coefficient de la solution sera ce dernier.

Le coefficient de disponibilité est une information qui est indiquée dans le Service Level Agreement (SLA) du fournisseur de services.

Dans la suite de cette section, nous allons voir que la disponibilité va dépendre des facteurs suivants : la fiabilité (reliability), la récupération (recovery), la facilité d'entretien (serviceability) et la gérabilité (manageability) [21].

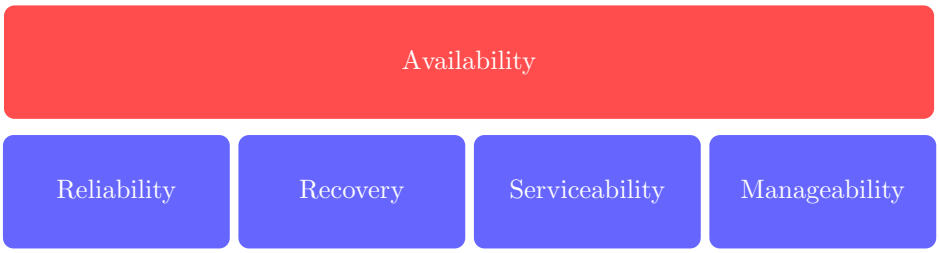


FIGURE 5.1 – Les éléments de la disponibilité [21]

5.1 Fiabilité (reliability)

L'Institute of Electrical and Electronics Engineers (IEEE) Reliability Society définit la fiabilité comme étant *"une discipline d'ingénierie de conception qui applique les connaissances scientifiques afin d'assurer qu'un produit va remplir sa fonction pour la durée requise dans un environnement donné"* [48]. Les outils principalement utilisés pour assurer la fiabilité d'un Cloud sont :

1. La redondance : la redondance va consister *"à disposer plusieurs exemplaires d'un même équipement ou d'un même processus ou de tout autre élément"* [56] dans l'infrastructure. Dans notre cas, cette redondance va s'effectuer à différents niveaux [25] :
   (a) Au niveau du serveur physique : par la redondance des cartes réseaux et de l'alimentation, par des disques en RAID, etc.
   (b) Au niveau logiciel : par l'utilisation de clusters, par la réplication des données, etc.
   (c) Au niveau du datacenter : par l'utilisation de zones multiples, qui séparent physiquements les centres les uns des autres.
   (d) Au niveau du Cloud Provider : par l'utilisation de plusieurs Cloud Provider distincts, pour se prémunir contre les fins d'activité des entreprises propriétaires.

   La redondance peut être de plusieurs types [52] :
   – Active/Cold standby : le site backup est démarré si nécessaire ;
   – Active/Hot standby : le site backup est démarré et prêt à prendre la main ;
   – Active/Active : le site backup est actif et traite des requêtes. Ce mode s'apparente dans ce cas à du load-balancing.

2. Le failover : le failover est la capacité de *"passer à un serveur, système, composant ou réseau redondant en cas de défaillance ou d'interruption anormale de l'application précédemment active"* [60]. Le failover est donc couplé à la redondance pour assurer la fiabilité des équipements.

Reliability	Cloud Provider	SaaS Provider
Physical redundancy	X	
Software redundancy	X	X
Datacenter redundancy	X	
Cloud Provider redundancy		X
Failover	X	X

5.2 Récupération (recovery)

La récupération est le retour à une situation normale après une interruption de services. Elle va consister en un ensemble de procédures prévoyant l'apparition de situations spécifiques et les actions à entreprendre pour rétablir le système à un état de fonctionnement normal. Les procédures vont concerner aussi bien les petits problèmes quotidiens (panne d'un serveur ou d'un routeur, etc) que les problèmes très importants (coupure d'électricité, catastrophe naturelle, etc).

Etant donné que toute la gestion de l'infrastructure matérielle repose sur le Cloud Provider, c'est lui qui sera en charge d'assurer la maintenance et l'évolution de ces procédures. Le SaaS Provider peut également prévoir ce type de procédures, qui va principalement consister à mettre en place un plan de secours en cas de défaillance du Cloud Provider.

Recovery	Cloud Provider	SaaS Provider
Recovery plans	X	
Recovery plans after Cloud Provider failure		X

5.3 Facilité d'entretien (serviceability)

La facilité d'entretien est *"la capacité de déterminer efficacement l'origine d'un problème, de le diagnostiquer et de le corriger"* [21]. Nous visons ici deux expertises :

– l'expertise du Cloud Provider, dont les équipes sont spécialisées dans la gestion de datacenters, ce qui implique un ensemble de compétences variées (software, hardware, électricité, air conditionné, réseau, etc). Cette expertise doit même dépasser le cadre du datacenter : la plupart des Cloud

Providers gèrent plusieurs sites et ont la capacité de basculer de l'un vers l'autre en cas de problème.

– l'expertise du SaaS Provider, qui doit pouvoir assurer la maintenance de son application et prendre les actions correctives nécessaires en cas d'interruption de service.

Le facilité d'entretien représente donc l'expertise technique que le Cloud Provider et le SaaS Provider apportent pour gérer les éléments dont ils ont la charge. Cette propriété est particulièrement importante, principalement du côté du Cloud Provider, étant donné que le SaaS Provider et l'utilisateur final (SaaS User) lui ont transféré la responsabilité de la gestion de leurs ressources informatiques, dont leurs activités dépendent crucialement.

Serviceability	Cloud Provider	SaaS Provider
(multiple) Data center(s) expertise	X	
Application expertise		X

5.4 Facilité de gestion (manageability)

La gérabilité est *"la capacité de créer et maintenir un environnement qui limite l'impact négatif que les personnes peuvent avoir sur le système"* [21]. Il s'agit donc de l'ensemble des outils qui permettent de limiter les actions et dégâts qu'une personne autorisée ou non peut effectuer sur le système. Nous retrouverons donc les mêmes outils que ceux mis en place pour l'isolation du chapitre sécurité, et en particulier la virtualisation, qui permet de séparer la couche matérielle de la couche logicielle et de scinder complètement les environnements des utilisateurs [41].

Manageability	Cloud Provider	SaaS Provider
Virtualization	X	X
Multitenancy	X	X
Security tools	X	X

5.5 Synthèse de la disponibilité

Le tableau suivant reprend l'ensemble des caractéristiques et outils liés à la disponibilité d'une solution Cloud. La symbolique utilisée est identique à celle utilisée dans le chapitre sécurité.

TABLE 5.2 – Tableau synthétique sur la répartition des outils de disponibilité dans une solution Cloud

Reliability	Outils	
	Physical redundancy	C
	Software redundancy	C|S
	Datacenter redundancy	C
	Cloud Provider redundancy	S
	Failover	C|S
Recoverability		
	Recovery plans	C
	Recovery plans after Cloud Provider failure	S
Serviceability		
	(multiple) Data center(s) expertise	C
	Application expertise	S
Manageability		
	Virtualization	C|S
	Multitenancy	C|S
	Security tools	C|S

Chapitre 6

Proposition d'un système d'évaluation et de comparaison

Pour permettre d'évaluer une solution de type SaaS, nous proposons l'utilisation d'un questionnaire dans lequel nous avons synthétisé le contenu des chapitres 3, 4 et 5. Cette méthode d'évaluation a été choisie car elle est simple d'utilisation, a un coût faible et donne un résultat immédiatement.

Ce questionnaire est composé de trois pages, correspondant aux différents chapitres : sécurité & vie privée, scalability et disponibilité.

Chaque page comporte une liste d'outils, éventuellement divisée en sections qui indiquent l'emplacement de ceux-ci : par exemple, le cryptage (encryption) est un outil de la page "Security & Privacy", qui est repris dans la section "Network level" et la section "Data". La personne souhaitant évaluer une solution de type SaaS doit alors indiquer la présence (Y) ou l'absence (N) de ces outils.

Notre démarche nous a amenés à produire un questionnaire fonctionnel sous la forme d'un tableur Excel, qui se présente ainsi :

Security			Existing (Y/N)
	Network level		
		Firewall	Y/N
		Network virtualization	Y/N
		Encryption	Y/N
		Tunneling	Y/N
		Route analytics	Y/N
		Network-level logs	Y/N
		IDS	Y/N
	Host level		
	Hypervisor	Events monitoring	Y/N
		Limit access	Y/N
		Limit threat surface	Y/N
	Instances	Events monitoring	Y/N
		Limit access	Y/N
		Limit threat surface	Y/N
		Antivirus	Y/N
		Firewall	Y/N
		IDS	Y/N
		Strong authentication	Y/N
	Application level		
		Sandbox architecture	Y/N
		Multitenancy	Y/N
		Virtualization	Y/N
		Monitoring	Y/N
		User authentication	Y/N
		Single sign-on	Y/N
		Privilege management	Y/N
		SSL or TLS	Y/N
		Secured APIs	Y/N
	Data		
		Contrôle d'accès	Y/N
		Encryption	Y/N
		MAC	Y/N
		Programmatic checks	Y/N
		Backup	Y/N
		Redondance	Y/N
Privacy			
		Collection limitation	Y/N
		Use limitation	Y/N
		Security	Y/N
		Retention	Y/N
		Destruction	Y/N
		Destruction via encryption	Y/N
		Transfer	Y/N
		Accountability	Y/N

Scalability			Existing (Y/N)
	Resources scaling		
		Add/remove instance resources	Y/N
		Add/remove new instances	Y/N
		Hot-swapping/Cold-swapping	Y/N
		Multiple configurations	Y/N
	Infrastructure software		
		Load balancing	Y/N
		Auto-scaling layer	Y/N
		Snapshots	Y/N
	Application		
		SOA / Minimal coupling	Y/N
		Application tuning	Y/N
		Multitenancy	Y/N
	Storage		
		SQL-based databases	Y/N
		NoSQL databases	Y/N
		Caching services	Y/N
		Multitenancy	Y/N
Availability			
		Physical redundancy	Y/N
		Software redundancy	Y/N
		Datacenter redundancy	Y/N
		Cloud Provider redundancy	Y/N
		Failover	Y/N
		Recovery plans	Y/N
		Recovery plans after Cloud Provider failure	Y/N
		(multiple) Data center(s) expertise	Y/N
		Application expertise	Y/N
		Virtualization	Y/N
		Multitenancy	Y/N
		Security tools	Y/N

La méthode de pondération que nous utilisons dans ce modèle est relativement simple : chaque page porte sur une série de sous-thèmes, pour lesquels nous additionnons les présences des outils en leur attribuant un poids uniforme. Cette somme par sous-thème est ensuite mise en proportion par rapport à la solution idéale, c'est-à-dire celle qui exploite l'ensemble des outils requis. Par exemple, l'isolation au niveau de la sécurité nécessite au niveau de l'application une architecture "sandbox", multitenante et monitorée, comme nous l'avons indiqué dans le tableau récapitulatif du chapitre 3. Si un seul outil sur les trois est présent, le score obtenu est donc de 33%.

Le pourcentage calculé est ensuite repris dans un graphe de type radar, structuré autour des sous-thèmes, comme illustré aux Figures 6.1, 6.2 et 6.3. Ce type de graphique présente l'avantage d'illustrer le niveau atteint dans chaque sous-thème et de comparer les valeurs obtenues entre elles.

La quatrième page de notre questionnaire fusionne les trois graphes radars des pages précédentes pour obtenir une vue synoptique de la solution SaaS, structurée par thèmes et sous-thèmes. Un exemple de ce graphique est indiqué à la Figure 6.4.

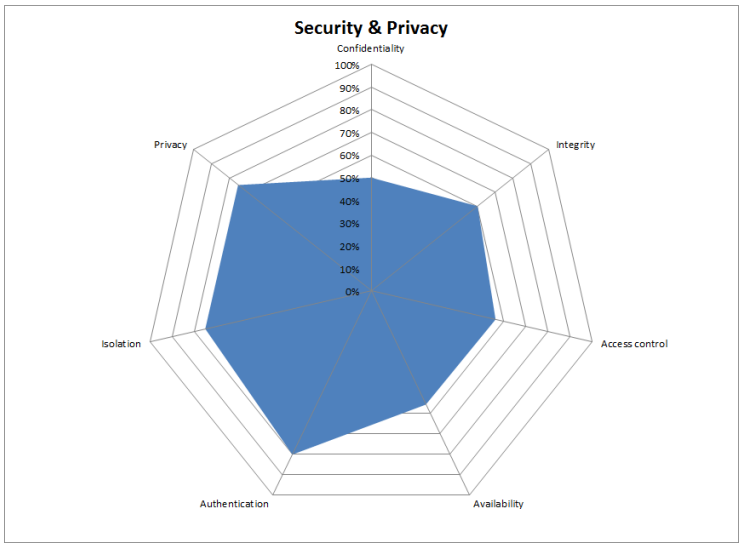


FIGURE 6.1 – L'évaluation de la sécurité et de la vie privée

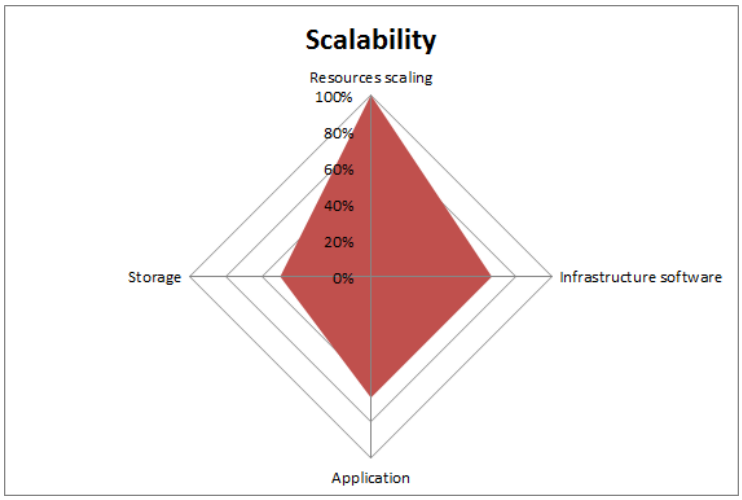


FIGURE 6.2 – L'évaluation de la scalability

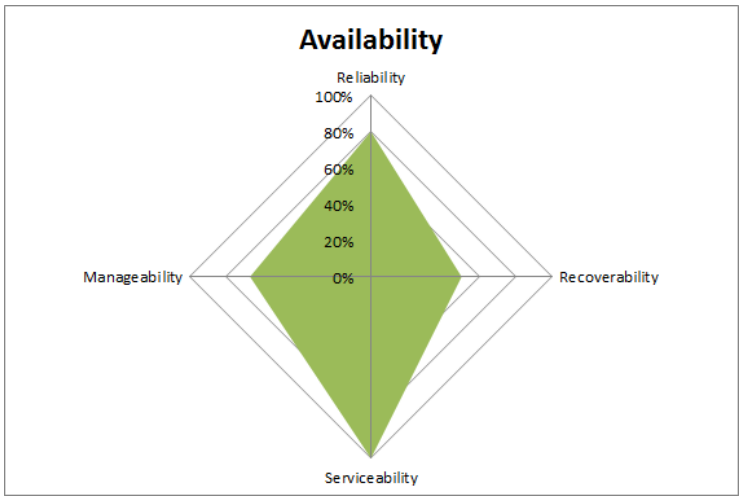


FIGURE 6.3 – L'évaluation de la disponibilité

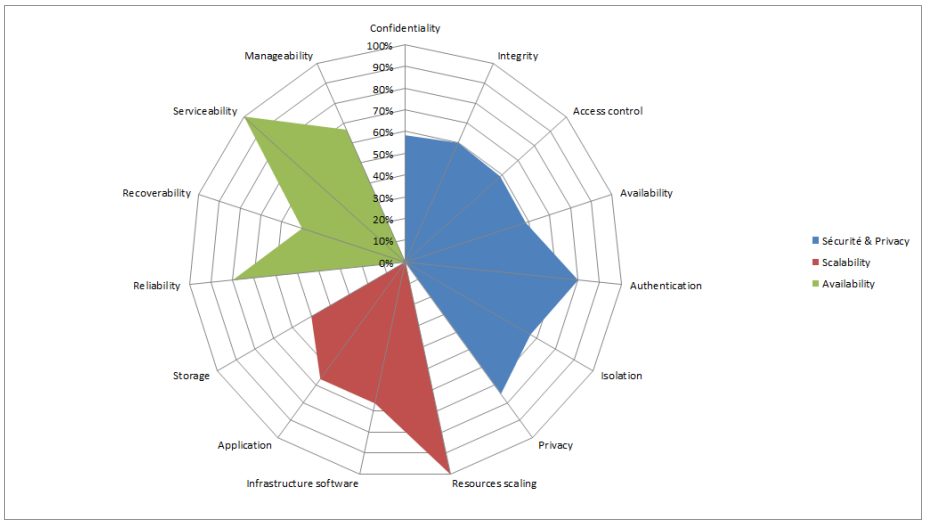


FIGURE 6.4 – Graphique synthétisant l'évalution de la solution de type SaaS

Exemple

Pour illustrer l'utilisation du questionnaire, nous proposons ici d'évaluer la solution Amazon Web Services d'un point de vue Cloud Provider. Nous nous plaçons donc du côté d'une entreprise souhaitant fournir une application en mode SaaS et qui est à la recherche d'un cloud pour assurer l'hébergement de son produit. L'évaluation doit permettre de déterminer le niveau de services d'AWS par rapport à un concurrent potentiel et également préciser quelles sont les fonctionnalités que l'entreprise doit implémenter dans sa solution pour atteindre un score acceptable dans chaque sous-thème vis-à-vis du SLA que l'entreprise a avec ses propres clients.
Nous limiterons ici notre étude d'AWS au thème sécurité. Cette limitation nous est imposée par le manque de temps et de ressources. Les réponses indiquées proviennent principalement des informations disponibles sur le site web du fournisseur et sur [5].

Security	Amazon Web Services		Existing (Y/N)
	Network level		
		Firewall	Y
		Network virtualization	Y
		Encryption	Y
		Tunneling	Y
		Route analytics	Y
		Network-level logs	Y
		IDS	Y
	Host level		
	Hypervisor	Events monitoring	Y
		Limit access	Y
		Limit threat surface	Y
	Instances	Events monitoring	N
		Limit access	N
		Limit threat surface	N
		Antivirus	N
		Firewall	Y
		IDS	N
		Strong authentication	N
	Application level		
		Sandbox architecture	N
		Multitenancy	N
		Virtualization	Y
		Monitoring	N
		User authentication	Y
		Single sign-on	N
		Privilege management	Y
		SSL or TLS	Y
		Secured APIs	Y
	Data		
		Contrôle d'accès	Y
		Encryption	Y
		MAC	N
		Programmatic checks	N
		Backup	Y
		Redondance	Y
Privacy			
		Collection limitation	N
		Use limitation	N
		Security	Y
		Retention	N
		Destruction	N
		Destruction via encryption	N
		Transfer	N
		Accountability	N

Sans surprise, la solution d'Amazon comprend pratiquement l'ensemble des fonctionnalités qu'un Cloud Provider peut pourvoir. Les fonctionnalités manquantes sont peu nombreuses, malgré le modèle de services IaaS du fournisseur : AWS propose par exemple des mécanismes d'authentification et de gestion des privilèges via son système AWS Identity and Access Management (IAM). Certaines fonctionnalités comme le single sign-on sont absentes mais sont prévues dans un futur proche [4]. Cette annonce tend à démontrer qu'Amazon poursuit l'extension de son offre et que son modèle de services actuel, l'IaaS, implémente des fonctionnalités qui se retrouvent généralement chez les fournisseurs de PaaS. Il est donc probable qu'à terme, Amazon soit en mesure d'offrir les deux types de services, que l'entreprise cliente pourra choisir en fonction de sa volonté de prendre en charge telle ou telle fonctionnalité.

La partie "vie privée" est plus difficile à évaluer : la collecte des données et l'usage de celles-ci est de la responsabilité du SaaS Provider, étant donné que c'est lui qui fournit l'application aux utilisateurs finaux. Les autres principes de la vie privée évoqués à la section 3.2 sont davantage orientés vers le Cloud Provider mais peu d'informations sont disponibles sur ce qui est mis en place. Par exemple, des API de destruction de données sont évidemment présentes mais elles consistent à supprimer le mapping vers ces données, en attendant que l'espace occupé soit réalloué. La destruction telle que nous l'avons définie plus haut n'est donc pas entièrement couverte.

Nous avons toutefois pris la liberté d'affirmer que le principe de sécurité était appliqué par Amazon.

Le résultat de cette évaluation sur la sécurité d'AWS se présente sous la forme du radar de la Figure 6.5.

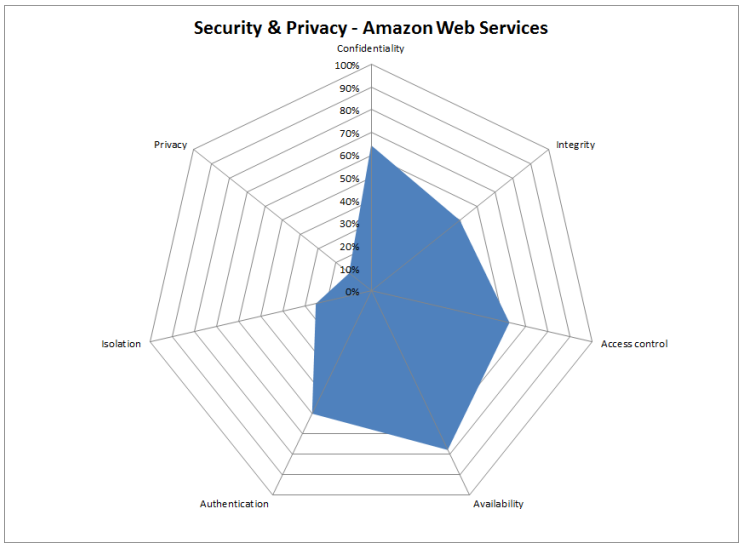


FIGURE 6.5 – Evaluation d'AWS sur le thème de la sécurité et de la vie privée

Discussion

Nous avons volontairement limité la précision de cette représentation en attribuant un poids identique à la présence de chaque outil. Une évolution possible de ce questionnaire serait d'affiner cette valeur pour obtenir un pourcentage plus précis de la couverture d'un sous-thème. Nous voyons deux possibilités pour la détermination de cette valeur :

- Attribuer un poids suivant son importance dans le sous-thème : il s'agit donc de donner une note de comparaison des outils entre eux, pour un sous-thème donné. Par exemple, l'évaluateur pourrait estimer que le cryptage du réseau est deux fois plus important qu'un système d'analyse du trafic (route analytics) pour assurer l'intégrité. La difficulté de cette méthode va venir du caractère subjectif de cette attribution du poids et donc de l'absence d'uniformité des évaluations faites par des évaluateurs différents.

- Attribuer un poids suivant la charge de travail et le coût que représente l'outil : cette configuration permettrait d'avoir une approche orientée "projet" dans la mise en place d'une solution de type SaaS. Elle présenterait l'avantage d'être relativement standard mais nécessiterait des mises à jour permanentes en fonction de l'évolution du prix des technologies et de leur complexité à être mises en oeuvre.

Nous avons également limité le type de réponse à une simple présence de l'outil (Y ou N), sans préciser si l'implémentation de l'outil est du ressort du Cloud Provider, du SaaS Provider ou des deux acteurs. Cette indication nous semble peu utile à ce niveau : l'utilisateur pourra faire référence aux tableaux récapitulatifs des chapitres 3, 4 et 5 pour déterminer l'entité responsable.

Utilisation

L'objectif de ce questionnaire est de constituer une première base de réflexion sur l'évaluation de solutions de type SaaS. Nous identifions plusieurs utilisations possibles de ce résultat :

- Il peut être utilisé pour comparer des solutions : ce questionnaire permet de réaliser une évaluation d'un fournisseur - qu'il soit Cloud Provider ou SaaS Provider - et de comparer le résultat obtenu avec un fournisseur équivalent ;

- Il peut servir de base à un Service Level Agreement (SLA) : les scores obtenus dans les sous-thèmes pourraient être indiqués dans ce type de contrat pour garantir la qualité du service. Le fournisseur serait alors libre de sélectionner les outils, pour autant que le résultat soit conforme aux

attentes du client. Toutefois, pour être complet, un aspect performance devra obligatoirement être ajouté, avec une série de métriques spécifiques ;

– Il peut servir de guide : un Cloud Provider ou SaaS Provider peut aisément évaluer son offre et débuter la mise en place de nouveaux outils en fonction des objectifs qu'il souhaite atteindre. Par exemple, s'il souhaite être davantage scalable, le questionnaire peut lui fournir la liste des outils conseillés pour augmenter son efficacité à ce niveau ;

– Il peut constituer la base d'une certification : celle-ci pourrait permettre de catégoriser les fournisseurs en fonction de leurs scores obtenus à ce test. Le choix du client en sera facilité et les fonctionnalités couvertes seront clairement exprimées.

Le résultat que nous proposons ici est une première ébauche théorique basée sur la littérature disponible. Le modèle de questionnaire constitue une première base d'évaluation technique et fonctionnelle d'une solution de type SaaS, qui devra être confrontée à la réalité et ajustée, avant d'être validée. D'autres aspects du Cloud Computing et du SaaS devront y être ajoutés : nous pensons notamment à la performance et à la portabilité de la solution, qui n'ont pas été abordées dans ce travail.

Pour permettre son exploitation en tant qu'outil d'aide à la décision d'une solution, une étude des modèles business utilisés par les fournisseurs de Cloud et de SaaS sera indispensable : elle permettra d'obtenir un ratio fonctionnalité/coût indiquant la solution la plus adaptée en fonction du budget défini par le SaaS Provider.

Chapitre 7

Conclusion

Dans cette étude sur le SaaS et le Cloud, nous avons commencé par définir le Cloud Computing, ses fonctionnalités, ses modèles de déploiement et de service. A partir de là, nous avons dégagé les deux rôles clés de ce type de solution : le SaaS Provider et le Cloud Provider.

Notre analyse de la littérature scientifique nous a permis d'identifier trois thèmes : la sécurité & la vie privée, la disponibilité et la scalability. Pour chacun d'entre eux, nous avons proposé une structure de décomposition en sous-thèmes et une série d'outils permettant de les implémenter. Pour chaque outil, nous avons indiqué quel acteur (SaaS Provider, Cloud Provider ou les deux) en a la charge. Nous avons ensuite synthétisé ces outils dans un tableau récapitulatif par thème.

Ces informations récoltées nous ont permis de proposer un questionnaire d'évaluation. Celui-ci se compose des trois thèmes - la sécurité & la vie privée, la disponibilité et la scalability - et permet d'indiquer la présence ou l'absence d'un outil (Y ou N). Un score est ensuite calculé pour chaque sous-thème et représenté dans un graphe de type radar, par thème puis global. Ce système de questionnaire a été retenu pour sa facilité d'utilisation, son faible coût et son retour immédiat.

Nous avons ensuite appliqué ce questionnaire à un exemple, la solution Amazon Web Services d'un point de vue Cloud Provider, sur le thème sécurité.

Dans notre discussion sur le questionnaire d'évaluation, nous avons abordé le système de pondération. Cette première version est simple : elle cumule la présence des outils par sous-thèmes, qu'elle met en proportion avec le nombre total d'outils. Deux approches ont été proposées pour affiner cette pondération : attribuer à l'outil un coefficient lié à son apport fonctionnel ou lié à son coût d'implémentation.
Notre discussion a également porté sur les utilisations de ce questionnaire. Nous

en avons identifié quatre : la comparaison de solutions, la base d'un SLA, la constitution d'un guide de bonnes pratiques et la base d'une certification.

En conclusion, nous sommes parvenus à proposer une première version théorique d'un système d'évaluation basé sur la littérature existante. Les thèmes traités - à savoir la sécurité & la vie privée, la scalability et la disponibilité - et le questionnaire développé devront être confrontés à la réalité, affinés en fonction des résultats et complétés suivant l'évolution des technologies et outils.

Nous distinguons également deux axes d'améliorations qui devront faire l'objet de travaux complémentaires : l'ajout de thèmes fonctionnels - comme la performance et la portabilité - et la prise en compte du modèle business sur lequel s'appuie le Cloud Computing. Nous estimons que la combinaison de ces différents travaux permettra à terme de définir la solution la plus adaptée à un projet, en tenant compte de ses caractéristiques techniques et de son coût.

Bibliographie

[1] Amazon. Amazon elastic compute cloud (amazon ec2). http://aws.amazon.com/fr/ec2/. [En ligne ; page disponible le 17 juillet 2013].

[2] Amazon. Aws auto scaling. http://aws.amazon.com/fr/autoscaling/. [En ligne ; page disponible le 19 juillet 2013].

[3] Amazon. Elastic load balancing. http://aws.amazon.com/fr/elasticloadbalancing/. [En ligne ; page disponible le 19 juillet 2013].

[4] Amazon. Faq aws identity and access management. http://aws.amazon.com/fr/iam/faqs/#Do_you_support_SAML_or_OAuth. [En ligne ; page disponible le 19 juillet 2013].

[5] Amazon. Livre blanc : Overview of security processes. http://media.amazonwebservices.com/pdf/AWS_Security_Whitepaper.pdf. [En ligne ; page disponible le 19 juillet 2013].

[6] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds : A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28, 2009.

[7] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4) :50–58, 2010.

[8] IEEE Standards Association. Standards in cloud computing. http://cloudcomputing.ieee.org/standards, 2013.

[9] Cor-Paul Bezemer and Andy Zaidman. Multi-tenant saas applications : maintenance dream or nightmare ? In *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*, IWPSE-EVOL '10, pages 88–92, New York, NY, USA, 2010. ACM.

[10] Darryl Chantry. Mapping applications to the cloud. http://msdn.microsoft.com/en-us/library/dd430340.aspx, 2009. [En ligne ; page disponible le 6 mai 2012].

[11] Chris Chenley. Hypervisors. http://blogs.technet.com/b/chenley/archive/2011/02/09/hypervisors.aspx, 2011. [En ligne ; page disponible le 16 juillet 2013].

[12] Terry Chia. Confidentiality, integrity, availability : The three components of the cia triad. http://security.blogoverflow.com/2012/08/

confidentiality-integrity-availability-the-three-components-of-the-cia-triad/, 2012. [En ligne ; page disponible le 16 juillet 2013].

[13] Jean-Noël Collin. Sécurité et fiabilité des systèmes informatiques (infom119), 2012 - 2013.

[14] NoSQL Database. Nosql definition. http://nosql-database.org/, 2009. [En ligne ; page disponible le 21 juillet 2013].

[15] Lydia Duijvestijn, Avin Fernandes, Pamela Isom, Dave Jewell, Martin Jowett, Elisabeth Stahl, and Todd R Stockslager. Performance implications of cloud computing. http://www-304.ibm.com/jct03001c/support/techdocs/atsmastr.nsf/fe582a1e48331b5585256de50062ae1c/50a7eda6d9a742be8625771f005f1742/$FILE/Cloud%20Performance%20050610.pdf, 2010.

[16] Renaud Edouard-Baraud. L'architecture 'multitenant'. http://pro.01net.com/editorial/339595/larchitecture-multitenant/, 2007. [En ligne ; page disponible le 23 janvier 2013].

[17] John Gantz and David Reinsel. Extracting value from chaos. *IDC iView*, pages 1–12, 2011.

[18] Jérôme Garay. Richard stallman : le cloud computing, un piège propriétaire. http://www.generation-nt.com/stallman-cloud-computing-logiciel-proprietaire-actualite-163041.html, 2008.

[19] Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29(4) :1012 – 1023, 2013. Special Section : Utility and Cloud Computing.

[20] J. Gray and D.P. Siewiorek. High-availability computer systems. *Computer*, 24(9) :39–48, 1991.

[21] Michael Hawkins and Floyd Piedad. *High Availability : Design, Techniques and Processes*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[22] John Hermans. From hype to future - kpmg's 2010 cloud computing survey. http://www.kpmg.com/nl/nl/issuesandinsights/articlespublications/pages/fromhypetofuture.aspx, 2010. [En ligne ; page disponible le 30 mai 2013].

[23] Mark D. Hill. What is scalability ? *SIGARCH Comput. Archit. News*, 18(4) :18–21, December 1990.

[24] IBM. Going hybrid : Best of both worlds in cloud computing. http://www-304.ibm.com/businesscenter/cpe/html0/228279.html, 2012.

[25] Brian Jimerson. Software architecture for high availability in the cloud. http://www.oracle.com/technetwork/articles/cloudcomp/jimerson-ha-arch-cloud-1669855.html, 2012. [En ligne ; page disponible le 22 juillet 2013].

[26] Mike Kavis. Scaling in the cloud – part 1 : Distributing the load. http://www.kavistechnology.com/blog/%E2%80%8Bscaling-in-the-cloud-part-1-distributing-the-load/, 2012. [En ligne ; page disponible le 19 juillet 2013].

[27] Diana Kelley. Cloud computing security : Infrastructure issues. http://searchcloudsecurity.techtarget.com/tip/Cloud-computing-security-Infrastructure-issues, 2009. [En ligne ; page disponible le 15 juillet 2013].

[28] C. Kopparapu. *Load Balancing Servers, Firewalls, and Caches*. Wiley, 2002.

[29] Bill Laing. Windows azure service disruption update. http://blogs.msdn.com/b/windowsazure/archive/2012/03/01/windows-azure-service-disruption-update.aspx, 2012.

[30] Ming Mao, Jie Li, and M. Humphrey. Cloud auto-scaling with deadline and budget constraints. In *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, pages 41–48, 2010.

[31] Chris Marsh. Data integrity in the cloud. http://www.wwpi.com/index.php?option=com_content&view=article&catid=99:cover-story&id=12800:data-integrity-in-the-cloud&Itemid=2701018, 2011. [En ligne ; page disponible le 16 juillet 2013].

[32] Tim Mather, Subra Kumaraswamy, and Shahed Latif. *Cloud Security and Privacy : An Enterprise Perspective on Risks and Compliance*. O'Reilly Media, Inc., 2009.

[33] Peter Mell and Timothy Grance. The nist definition of cloud computing. Technical report, National Institute of Standards and Technology, 2011.

[34] Microsoft. Privacy in the cloud. http://www.microsoft.com/privacy/cloudcomputing.aspx, 2010.

[35] American Institute of Certified Public Accountants. Generally accepted privacy principles. http://www.aicpa.org/InterestAreas/InformationTechnology/Resources/Privacy/GenerallyAcceptedPrivacyPrinciples/Pages/default.aspx, 2009.

[36] Donn Parker. Principles of information security. http://www.informationintegrity.org/principles-of-information-security/, 2002. [En ligne ; page disponible le 16 juillet 2013].

[37] Tim Perdue. Nosql : An overview of nosql databases. http://newtech.about.com/od/databasemanagement/a/Nosql.htm, 2012. [En ligne ; page disponible le 21 juillet 2013].

[38] Thierry Janvier Philippe Grange and Francis Behr. Le livre blanc du cloud computing. Technical report, Syntec Informatique, 2010.

[39] Christian Plattner, Gustavo Alonso, and M. Tamer Özsu. Dbfarm : A scalable cluster for multiple databases. In *In Middleware*, pages 180–200, 2006.

[40] Office québécois de la langue française. Scalability. http://www.gdt.oqlf.gouv.qc.ca/resultat.aspx?terme=scalability, 2013. [En ligne ; page disponible le 3 août 2013].

[41] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pages 44–51. Ieee, 2009.

[42] Jeanne W Ross and George Westerman. Preparing for utility computing : The role of it architecture and relationship management. *IBM systems journal*, 43(1) :5–19, 2004.

[43] Margaret Rouse. authentication. http://searchsecurity.techtarget.com/definition/authentication, 2007. [En ligne ; page disponible le 16 juillet 2013].

[44] Simon Rowland. Horizontal vs vertical scaling. http://singinghorsestudio.com/horizontal-vs-vertical-scaling/, 2012. [En ligne ; page disponible le 19 juillet 2013].

[45] SalesForce. The force.com multitenant architecture white paper. Technical report, Force.com, 2008.

[46] IBM Global Services. Improving systems availability. http://www.cs.cmu.edu/~priya/hawht.pdf. [En ligne ; page disponible le 21 juillet 2013].

[47] Dave Shackleford. Securing the hypervisor : expert tips. http://www.computerweekly.com/opinion/Securing-the-hypervisor-expert-tips, 2012. [En ligne ; page disponible le 16 juillet 2013].

[48] IEEE Reliability Society. Ieee reliability society - reliability engineering. http://rs.ieee.org/about-rs.html. [En ligne ; page disponible le 22 juillet 2013].

[49] Elisabeth Stahl, Andrea Corona, Frank De Gilio, Marcello Demuro, Ann Dowling, Lydia Duijvestijn, Avin Fernandes, Dave Jewell, Bharathraj Keshavamurthy, Shmuel Markovits, Chandrakandh Mouleeswaran, Shawn Raess, and Kevin Yu. Performance and capacity themes for cloud computing. http://www.redbooks.ibm.com/redpapers/pdfs/redp4876.pdf, 2013.

[50] Computing Cloud Storage. Cloud service provider. http://www.computingcloudstorage.com/cloud-service-provider/. [En ligne ; page disponible le 22 juillet 2013].

[51] Reiji Suda, Ken Naono, Keita Teranishi, and John Cavazos. Software automatic tuning : Concepts and state-of-the-art results. In Ken Naono, Keita Teranishi, John Cavazos, and Reiji Suda, editors, *Software Automatic Tuning*, pages 3–15. Springer New York, 2010.

[52] Greg Thompson. High scalability and reliability in the cloud. http://www.slideshare.net/gmthomps/scalability-and-reliability-in-the-cloud. [En ligne ; page disponible le 22 juillet 2013].

[53] Luis M. Vaquero, Luis Rodero-Merino, and Rajkumar Buyya. Dynamically scaling applications in the cloud. *SIGCOMM Comput. Commun. Rev.*, 41(1) :45–52, January 2011.

[54] Clément Vouillon. Iaas, paas, saas : définition des 3 modèles de service du cloud. http://mag.welovesaas.com/index.php/2012/iaas-paas-saas-definitions-des-3-modeles-de-service-du-cloud/, 2012.

[55] Wikipédia. Confidentialité — wikipédia, l'encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=Confidentialit%C3%A9&oldid=94388789, 2013. [En ligne ; page disponible le 16-juillet-2013].

[56] Wikipédia. Redondance des matériels — wikipédia, l'encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=Redondance_des_

mat%C3%A9riels&oldid=92632040, 2013. [En ligne ; page disponible le 22 juillet 2013].

[57] Wikipédia. Sécurité des systèmes d'information — wikipédia, l'encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=S%C3%A9curit%C3%A9_des_syst%C3%A8mes_d%27information&oldid=95016645, 2013. [En ligne ; page disponible le 23 juillet 2013].

[58] Wikipedia. Cloud computing — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Cloud_computing&oldid=479244245, 2012. [En ligne ; page disponible le 20 février 2012].

[59] Wikipedia. Multitenancy — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Multitenancy&oldid=521455557, 2012. [En ligne ; page disponible le 23 janvier 2013].

[60] Wikipedia. Failover — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Failover&oldid=540839541, 2013. [En ligne ; page disponible le 22 juillet 2013].

[61] Wikipedia. Snapshot (computer storage) — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Snapshot_(computer_storage)&oldid=542542670, 2013. [En ligne ; page disponible le 19 juillet 2013].